Arquitetura Escalável de Processamento de Eventos Complexos para Geração Automatizada de Alertas Fiscais

Pedro Medeiros[1], Marcelo Iury S. Oliveira[1], Telmo Menezes[2], Raoni Kulesza[1],

Thais Gaudêncio[1], Yuri Malheiros[1]

[1]Universidade Federal da Paraíba

[2]University of Bristol

pedraugvsto@gmail.com, telmo.silvafilho@bristol.ac.ukv,

[marcelo, raoni, gaudencio.thais, yuri]@ci.ufpb.br

Abstract. *The digitalization of public administration drives data-driven solutions to enhance public policy efficiency. In this context, goods inspection faces the challenge of managing large and complex data flows. This work presents the development of a complex event processing system for automated tax alert generation. Using an event-driven architecture with Apache Kafka, parallel processing in Python, and real-time notifications, the solution correlates fiscal documents and vehicle records to identify irregularities. Tested in an operational environment, the system processed over 7,000 events per minute, strengthening data-driven tax inspection practices.*

Resumo. *A digitalização da administração pública impulsiona soluções baseadas em dados para tornar as políticas públicas mais eficientes. Neste contexto, a fiscalização de mercadorias lida com o desafio de enfrentar um grande e complexo fluxo de dados. Este trabalho apresenta a construção de um processador de eventos complexos para geração de alertas fiscais automatizados. Utilizando arquitetura orientada a eventos com Apache Kafka, processamento paralelo em Python e notificações em tempo real, a solução correlaciona documentos fiscais e registros veiculares para identificar irregularidades. Testado em ambiente operacional, ele foi capaz de processar pouco mais de 7000 eventos por minuto, fortalecendo a fiscalização tributária.*

1. Introdução

Nos últimos anos, a transformação digital da administração pública tem impulsionado a adoção de tecnologias capazes de apoiar a tomada de decisão baseada em dados. A crescente disponibilidade de dados governamentais, aliada ao avanço das infraestruturas de processamento distribuído, tem permitido o desenvolvimento de sistemas capazes de transformar grandes volumes de dados operacionais em informações estratégicas para gestão pública e fiscalização (Morabito, 2015; Vasilopoulou et al., 2023). Nesse contexto, abordagens baseadas em análise de dados e processamento em larga escala têm sido cada vez mais exploradas para melhorar a eficiência, a transparência e a capacidade analítica das instituições públicas.

A fiscalização do transporte de mercadorias representa um domínio particularmente desafiador nesse cenário. O crescimento do comércio eletrônico, a digitalização de documentos fiscais e a expansão das cadeias logísticas ampliaram significativamente o volume e a complexidade dos dados gerados nas operações de transporte. Como consequência, métodos tradicionais de fiscalização, frequentemente baseados em inspeções manuais ou análises retrospectivas de dados, apresentam limitações para lidar com fluxos massivos de informações geradas continuamente por diferentes sistemas e sensores.

Uma abordagem promissora para lidar com esse tipo de cenário é o paradigma de Processamento de Eventos Complexos (*Complex Event Processing* – CEP). CEP permite analisar fluxos contínuos de eventos provenientes de múltiplas fontes e identificar padrões relevantes em tempo real (Luckham, 2002). Em sistemas orientados a CEP, eventos simples são correlacionados por um motor de processamento capaz de aplicar regras e detectar condições específicas de interesse operacional ou de risco. De acordo com Kyrama e Gounaris (2025), arquiteturas modernas de CEP combinam técnicas de processamento distribuído e mensageria para viabilizar a análise escalável de eventos em ambientes de alta vazão de dados. Nesse contexto, a inteligência do sistema reside no mecanismo de correlação de eventos, análise temporal e detecção de padrões, que permite transformar fluxos de dados brutos em eventos complexos capazes de apoiar processos de decisão.

Este trabalho apresenta o desenvolvimento do BuiltinWarningConfigManager (BWCM), um processador de eventos complexos integrado a uma plataforma de monitoramento fiscal utilizada pela Secretaria de Estado da Fazenda da Paraíba . O sistema foi projetado para correlacionar eventos provenientes de emissões de documentos fiscais eletrônicos e registros de passagens veiculares capturados por câmeras com reconhecimento óptico de caracteres (OCR). A partir dessa correlação, o processador aplica regras fiscais para identificar situações potencialmente irregulares e gerar alertas automatizados que auxiliam as equipes de fiscalização.

As principais contribuições deste trabalho são:
(i) a proposição de uma arquitetura escalável de processamento de eventos complexos voltada ao monitoramento fiscal;
(ii) a implementação do processador BWCM baseado no padrão de processamento distribuído Manager/Worker; e
(iii) a avaliação do sistema em ambiente operacional com dados fiscais reais, demonstrando sua capacidade de processar grandes volumes de eventos e apoiar atividades de fiscalização orientadas por dados.

Dessa forma, a proposta contribui para a agenda de governo orientado por dados ao demonstrar como arquiteturas de processamento de eventos podem apoiar práticas de fiscalização tributária mais eficientes e baseadas em evidências.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados ao uso de CEP e análise de dados em larga escala. A Seção 3 descreve a arquitetura do sistema BWCM, incluindo seus principais componentes e o

pipeline de processamento de eventos. A Seção 4 apresenta a metodologia de validação utilizada para avaliar a arquitetura proposta. A Seção 5 discute os resultados obtidos a partir da avaliação do sistema. Por fim, a Seção 6 apresenta as conclusões do trabalho e aponta possíveis direções para pesquisas futuras.

2. Trabalhos Relacionados

Com a evolução das arquiteturas orientadas a eventos e das plataformas de *streaming* de dados, CEP passou a ser integrado a infraestruturas de processamento distribuído capazes de lidar com volumes massivos de eventos em tempo real. Kyrama e Gounaris (2025) discutem o estado atual das arquiteturas de CEP e destacam sua importância em ambientes distribuídos e em cenários de *edge computing*, nos quais a análise de eventos deve ocorrer com baixa latência e alta escalabilidade. Nesse contexto, tecnologias como Apache Kafka e *frameworks* de processamento distribuído têm sido amplamente adotadas para implementar *pipelines* de eventos capazes de suportar aplicações críticas.

No domínio da administração pública, o uso de análise de dados e arquiteturas de processamento em larga escala tem se tornado um elemento central da transformação digital do setor público. Morabito (2015) argumenta que a utilização de *big data* e *analytics* no governo permite transformar grandes volumes de dados administrativos em inteligência estratégica para formulação e monitoramento de políticas públicas. De forma complementar, Vasilopoulou et al. (2023) destacam que técnicas avançadas de análise de dados têm desempenhado papel fundamental na modernização de serviços públicos e na criação de sistemas de governo orientados por dados.

Diferentes aplicações de tecnologias digitais no setor público vêm sendo implementadas, como por Pedrosa e Canedo (2025), que investigam métodos de avaliação de serviços digitais governamentais no Brasil, destacando desafios relacionados à qualidade e à efetividade das plataformas digitais utilizadas pela administração pública. De forma semelhante, Silva et al. (2025) discutem práticas de garantia de qualidade aplicadas ao sistema CACTO, uma plataforma voltada ao apoio da fiscalização tributária no estado da Paraíba.

Apesar desses avanços, ainda existem desafios relacionados à integração de fluxos massivos de dados fiscais e logísticos em sistemas capazes de identificar automaticamente padrões de risco em tempo real. Nesse cenário, o presente trabalho contribui ao propor uma arquitetura de processamento de eventos complexos aplicada ao monitoramento fiscal, explorando técnicas de mensageria distribuída, processamento paralelo e análise baseada em regras para apoiar atividades de fiscalização tributária orientadas por dados.

3. Arquitetura do Sistema

Esta seção descreve a arquitetura e a implementação do sistema proposto, destacando os principais componentes responsáveis pela ingestão, processamento e geração de alertas

a partir dos eventos fiscais. A Seção 3.1 apresenta uma visão geral da arquitetura do sistema. A Seção 3.2 descreve o fluxo de processamento dos eventos. A Seção 3.3 detalha o modelo de processamento baseado no padrão Manager/Worker. Por fim, a Seção 3.4 apresenta a arquitetura modular das regras de alerta e o mecanismo de geração de notificações fiscais.

3.1 Visão Geral

O BWCM apresentado neste trabalho foi projetado como um componente de *backend* de uma plataforma de monitoramento fiscal orientada a eventos. Seu principal objetivo é transformar fluxos contínuos de dados operacionais em alertas fiscais que auxiliem a atuação das equipes de fiscalização. O BWCM combina técnicas de mensageria distribuída, processamento paralelo e análise baseada em regras para identificar situações potencialmente irregulares a partir da correlação entre eventos de diferentes fontes de dados.

No que tange à segurança e ao controle de acesso, a arquitetura é executada integralmente em uma rede privada sob infraestrutura *on-premise*. A integração de dados é protegida por uma camada de API via Hasura, que garante que os usuários não possuam controle direto sobre as *queries* executadas pelo motor.

Entre as principais fontes de dados integradas pelo sistema destacam-se:

- eventos de emissão de documentos fiscais eletrônicos: Nota Fiscal Eletrônica (NF-e), Conhecimento de Transporte Eletrônico (CT-e) e Manifesto de Documento Fiscal Eletrônico (MDF-e);
- registros de passagens de veículos capturados por câmeras com reconhecimento de placas;
- bases de dados institucionais relacionadas a cadastro de contribuintes e operações fiscais.

Esses eventos são ingeridos e distribuídos por meio de uma infraestrutura de mensageria baseada em Apache Kafka, permitindo o processamento assíncrono e escalável dos dados.

3.2 *Pipeline* de Processamento de Eventos

O processamento dos eventos ocorre por meio de um *pipeline* estruturado composto pelas seguintes etapas:

1. Ingestão de eventos
2. Orquestração e enfileiramento
3. Processamento paralelo
4. Avaliação de regras fiscais
5. Persistência e notificação

A Figura 1 apresenta a arquitetura geral do processador, ilustrando o *pipeline* desde a ingestão de um evento até a geração do alerta. A arquitetura do BWCM foi projetada como um processamento assíncrono composto por componentes desacoplados, cada um responsável por uma etapa específica do fluxo de dados. Inicialmente, os eventos são ingeridos por meio de uma infraestrutura de *streaming* que atua como barramento de comunicação e garante ingestão escalável e resiliente dos dados. Em seguida, os dados são distribuídos para uma fila interna, que é consumida seguindo o padrão Manager/Worker (detalhado na Seção 3.3) para garantir a execução paralela em Python.

Nessa etapa, os Workers assumem o papel central de execução: cada evento padronizado é confrontado, regra a regra, com todas as configurações fiscais ativas. Esse processamento paralelo garante que as regras sejam aplicadas de forma modular e independente sobre os dados recebidos. O diferencial aqui é a capacidade de correlação hierárquica de eventos (CEP). O sistema utiliza a placa capturada para realizar um encadeamento (chaining), recuperando documentos de transporte (MDF-e ou CT-e) e as respectivas Notas Fiscais (NF-e) vinculadas em tempo real. Essa abordagem transforma o evento isolado em um contexto operacional rico, utilizando o objeto Document para garantir a fidedignidade entre as diferentes fontes de dados.

Após a avaliação das regras (aprofundada na Seção 3.4), o sistema realiza a persistência e distribuição do alerta por meio de uma camada de armazenamento dual. Os alertas são registrados em um banco de dados relacional PostgreSQL para rastreabilidade histórica. Paralelamente, as notificações são publicadas em *streams* via Redis, possibilitando a atualização em tempo real das interfaces das equipes de fiscalização. Essa integração permite que o BWCM opere com baixa latência e alta disponibilidade, mesmo diante de volumes massivos de eventos.

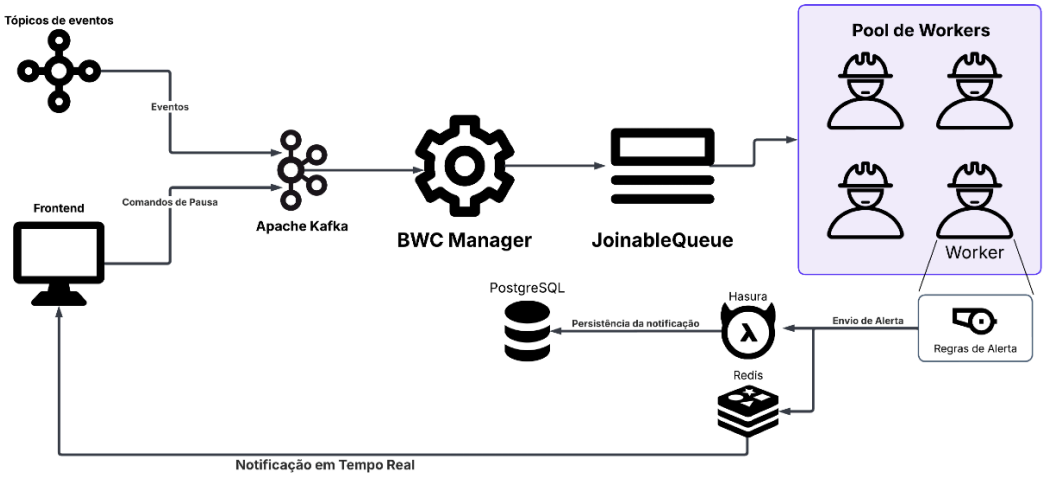

Figura 1. Diagrama de Arquitetura do BWCM. O diagrama apresenta o fluxo de processamento de eventos do sistema. Eventos provenientes de diferentes fontes são publicados em tópicos do Apache Kafka, que atua como barramento de mensageria.

Essa arquitetura, que combina mensageria assíncrona, processamento paralelo e uma camada de dados dual, permite que o BWCM opere com baixa latência, alta disponibilidade e escalabilidade horizontal, garantindo a confiabilidade do processamento, mesmo diante de um volume massivo de eventos. Em resumo, a linguagem Python, bem como pacote como Pandas foram usados para a implementação das regras de análise fiscal. O Apache Kafka foi usado para mensageria distribuída e o PostgreSQL para armazenamento persistente. O Hasura GraphQL foi usado como uma camada de API instantânea que se conecta ao PostgreSQL que compila cada consulta GraphQL em uma única query SQL otimizada. Por fim, Redis foi usado para comunicação em tempo real, bem como cache de dados. Essa integração permite construir uma arquitetura escalável e resiliente, capaz de transformar grandes volumes de dados fiscais em alertas operacionais que apoiam a fiscalização tributária.

3.3 O Padrão Manager/Worker

No BWCM, o processamento dos eventos é realizado por meio de um modelo distribuído baseado no padrão Manager/Worker. O componente Manager é responsável por consumir eventos dos tópicos Kafka, realizar verificações preliminares e encaminhar as mensagens para uma fila compartilhada bloqueante. Os Workers, por sua vez, executam o processamento paralelo dos eventos, padronizando os dados recebidos e avaliando as regras fiscais configuradas. Tanto o Manager quanto os Workers foram implementados em Python, com pacotes do estado da arte em processamento de dados, por exemplo Pandas[1], NumPy[2] e Ray[3]. Essa abordagem permite lidar eficientemente com grandes volumes de eventos provenientes de diferentes fontes.

3.4 Arquitetura Modular de Alertas

A lógica de detecção de irregularidades é implementada por meio de uma arquitetura modular de regras com uso de herança e polimorfismo. Como apresentado na Figura 2, a classe base BuiltInWarningConfigs é uma classe abstrata que define os métodos isSuitable, isRelevant e getInsertObject. Os dois primeiros determinam se (i) a regra é compatível com um tipo de evento e se (ii) o evento deve gerar um alerta, respectivamente. O último método retorna os dados a serem registrados de forma persistente.

Cada regra fiscal é implementada por meio da extensão da classe BuiltInWarningConfigs, permitindo a implementação de diferentes estratégias de detecção sem comprometer a estrutura do sistema. Um exemplo prático é a regra de "Contribuinte em Situação Irregular" (Tipo 108), ilustrada na Figura 2, que pode identificar, por exemplo, notas fiscais emitidas por CPFs de pessoas falecidas, ou no CNPJ de empresas baixadas.

[1] https://pandas.pydata.org/
[2] https://numpy.org/
[3] https://docs.ray.io/en/latest/index.html

De uma forma geral, as implementações de BuiltInWarningConfigs, durante a execução do método isRelevant, processam cada evento em múltiplas etapas, incluindo a verificação da aplicabilidade da regra, a análise das condições de risco e a geração do alerta correspondente quando as condições são satisfeitas.

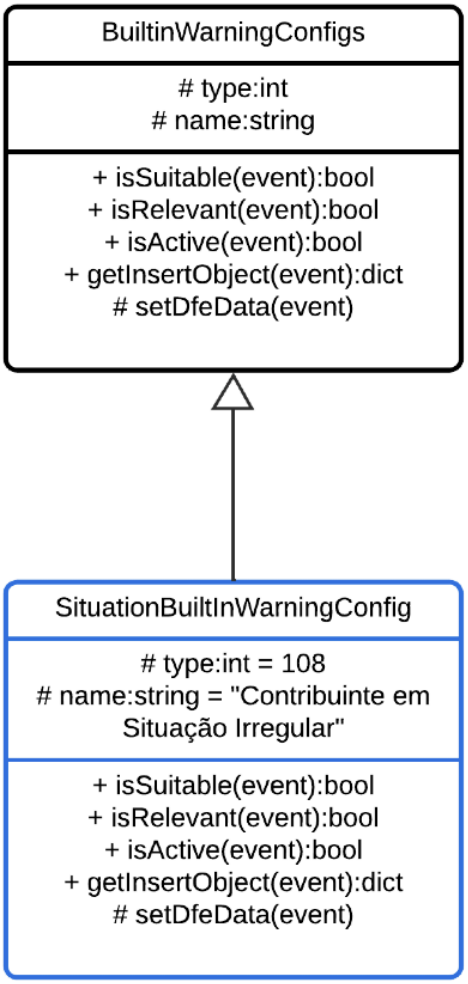

Figura 2. Diagrama de Classes do Módulo de Alerta. O diagrama de classes apresenta a arquitetura orientada a objetos utilizada para implementar as regras de detecção no BWCM.

4. Metodologia de Validação da Arquitetura

A avaliação da arquitetura proposta foi conduzida com o objetivo de verificar se o processador de eventos BWCM atende aos requisitos definidos e opera de forma

eficiente sob alta demanda. O processo de validação contemplou aspectos funcionais e não funcionais em um ambiente representativo de uso real.

Para a realização dos experimentos, utilizou-se um ambiente de desenvolvimento que replica fielmente a infraestrutura e o tráfego recebidos na plataforma à qual o BWCM se integra. A fim de simular cenários de carga elevada, foi desenvolvido um componente *producer* no Apache Kafka dedicado à geração de eventos sintéticos. Esses eventos foram modelados para serem análogos aos dados reais em termos de volume, frequência e complexidade semântica, permitindo validar a escalabilidade do sistema sem comprometer dados sensíveis em fase de teste.

Quanto à coleta de registros de passagem de veículos, cabe ressaltar que o BWCM atua como consumidor de fluxos de dados provenientes de câmeras operadas por entes governamentais. Tais dispositivos operam em regime 24/7 e utilizam iluminação infravermelha, o que garante a estabilidade da captura independentemente de condições climáticas (chuva) ou períodos de baixa luminosidade (noite). Uma vez que o BWCM é um sistema de processamento de eventos pós-ingestão, a responsabilidade pela acurácia da captura reside na infraestrutura de sensores de campo, focando o processador exclusivamente na correlação e na lógica fiscal avançada.

Inicialmente, foram avaliados os requisitos funcionais, verificando se os principais componentes da arquitetura, incluindo ingestão de eventos, processamento paralelo, avaliação de regras e geração de alertas, operam conforme as especificações do sistema. Essa etapa buscou garantir que o *pipeline* de processamento de eventos estivesse corretamente integrado e que os mecanismos de detecção de irregularidades fiscais fossem executados de forma adequada.

Em seguida, foram analisados os requisitos não funcionais, com foco em desempenho, escalabilidade e capacidade de processamento da arquitetura. Para isso, foram observados indicadores relacionados à vazão de eventos processados e ao comportamento do sistema sob carga elevada, considerando fluxos de dados provenientes de documentos fiscais eletrônicos e registros de passagens veiculares.

Além da análise quantitativa de desempenho, também foi realizada uma avaliação qualitativa da arquitetura, considerando aspectos relacionados à extensibilidade, modularidade e flexibilidade operacional do sistema. Essa análise buscou verificar a capacidade da arquitetura de evoluir ao longo do tempo, permitindo a inclusão de novas regras fiscais e adaptações às necessidades da administração tributária.

Por fim, foi conduzida uma análise dos benefícios operacionais proporcionados pelo sistema, considerando seu impacto no processo de fiscalização e no uso de dados fiscais para suporte à tomada de decisão. Essa etapa permitiu identificar como a arquitetura proposta contribui para ampliar a capacidade analítica das equipes de fiscalização e para fortalecer práticas de monitoramento fiscal orientadas por dados.

Os resultados obtidos a partir desse processo de validação são apresentados e discutidos na seção seguinte.

5. Resultados

Esta seção apresenta os resultados obtidos com a implementação do processador de eventos complexos BWCM, enfatizando a validação das funcionalidades definidas no projeto, o desempenho da arquitetura proposta e os benefícios operacionais observados na utilização do sistema. A análise foi conduzida com base na verificação dos requisitos funcionais e não funcionais definidos durante o desenvolvimento, além da observação do comportamento do sistema em execução.

5.1 Validação dos requisitos funcionais

A validação funcional do sistema teve como objetivo verificar se os principais componentes arquiteturais do BWCM atendem às especificações estabelecidas durante o projeto. Para isso, foram analisados os requisitos relacionados à ingestão de eventos, processamento paralelo, modularidade das regras de alerta e geração automatizada de notificações fiscais.

A avaliação indicou que todos os requisitos funcionais foram implementados e testados com sucesso. O sistema demonstrou capacidade de consumir eventos provenientes do barramento de mensagens, padronizar os dados recebidos, aplicar regras de negócio modulares e gerar alertas fiscais automaticamente quando condições de risco são identificadas.

Em particular, a validação confirmou:

- o funcionamento correto do mecanismo de consumo e enfileiramento de eventos, no qual o componente Manager recebe eventos do Kafka e os distribui para a fila compartilhada utilizada pelos workers;
- a padronização das mensagens em um modelo de dados interno denominado *Document*, permitindo tratamento uniforme de diferentes tipos de eventos;
- a implementação de uma arquitetura modular de regras de alerta, que permite adicionar novos tipos de detecção sem impactar as regras existentes;
- a execução do processo de avaliação em duas etapas (*isSuitable* e *isRelevant*), que permite filtrar eventos antes da aplicação da lógica completa de detecção;
- a geração automática de alertas quando eventos satisfazem as condições definidas pelas regras fiscais implementadas.

A validação sistemática desses requisitos fornece evidências de que a arquitetura proposta é capaz de implementar um pipeline completo de processamento de eventos orientado à geração automatizada de alertas fiscais.

5.2 Validação dos requisitos não funcionais

Além da verificação das funcionalidades do sistema, também foram avaliados aspectos relacionados ao desempenho, escalabilidade e capacidade de processamento da arquitetura.

Os experimentos indicam que o processador de eventos é capaz de lidar com fluxos de dados de alta vazão, mantendo desempenho consistente mesmo em cenários de carga elevada. Em ambiente de execução representativo do contexto operacional da plataforma, o sistema demonstrou capacidade de processar aproximadamente:

- 5.000 eventos de passagens veiculares por minuto, provenientes de câmeras com reconhecimento de placas;
- 2.200 emissões de documentos fiscais eletrônicos por minuto.

Como trata-se de um sistema de processamento de dados em tempo real, além do atendimento da vazão, outra avaliação importante é a média do tempo de processamento. O tempo de processamento médio dos eventos foi medido em cerca 300ms. No pior caso, os eventos eram tratados em até 1s. Desta forma, o BWCM também demonstrou capacidade de operar com baixa latência na geração de alertas, possibilitando que eventos detectados sejam rapidamente disponibilizados para análise pelas equipes de fiscalização.

5.3 Análise qualitativa da arquitetura

Além dos resultados quantitativos relacionados ao desempenho, a análise qualitativa da arquitetura evidencia propriedades importantes para a evolução do sistema no contexto de plataformas de governo digital.

Primeiramente, a adoção de uma arquitetura modular baseada em classes de regras permite que novas estratégias de detecção sejam incorporadas ao sistema de forma incremental. Essa característica é particularmente relevante em cenários de fiscalização tributária, nos quais novos padrões de evasão fiscal podem surgir ao longo do tempo.

Quanto à fidedignidade das detecções, vale salientar que o BWCM utiliza lógica determinística, o que minimiza a ocorrência de falsos positivos técnicos. O sistema atua como uma ferramenta de suporte à decisão, na qual os alertas servem como sugestões fundamentadas para triagem manual, otimizando a capacidade analítica sem substituir o julgamento final do auditor.

Outro aspecto relevante é a flexibilidade operacional do sistema. O mecanismo de gerenciamento dinâmico de regras permite ativar ou desativar configurações de alerta sem interromper a execução do sistema, o que facilita ajustes operacionais em ambientes de produção.

Além disso, a integração com tecnologias como PostgreSQL, Redis e Hasura contribui para garantir persistência confiável dos alertas gerados e distribuição eficiente das notificações para os sistemas de monitoramento utilizados pelos auditores fiscais.

5.4 Benefícios operacionais

Os resultados obtidos indicam que o BWCM contribui para aprimorar significativamente a capacidade de monitoramento fiscal baseada em dados.

Entre os principais benefícios observados destacam-se:

- automatização da detecção de irregularidades fiscais, reduzindo a dependência de análise manual;
- maior agilidade na identificação de eventos de risco, permitindo atuação mais rápida das equipes de fiscalização;
- melhor aproveitamento dos dados fiscais disponíveis, transformando fluxos massivos de eventos em inteligência operacional;
- capacidade de evolução contínua do sistema, graças à arquitetura modular de regras.

Essas características demonstram o potencial da solução como componente central de plataformas de fiscalização tributária orientadas por dados, alinhando-se às iniciativas contemporâneas de transformação digital da administração pública.

Cabe destacar também que, após a fase de validação em ambiente de testes, o sistema foi transposto para o ambiente de produção, onde permanece em operação contínua por mais de um ano, o que atesta a maturidade e a estabilidade da arquitetura diante do tráfego real.

6. Conclusões

Este trabalho apresentou o projeto, desenvolvimento e análise do BWCM, um processador de eventos complexos voltado à geração automatizada de alertas fiscais. A proposta explorou o uso de uma arquitetura distribuída baseada no padrão Manager/Worker, aliada à padronização semântica dos eventos por meio do objeto *Document* e a uma estrutura modular de regras de negócio. Essa combinação permitiu a construção de um pipeline de processamento capaz de correlacionar eventos fiscais e de tráfego em tempo real, gerando alertas que apoiam a atuação das equipes de fiscalização.

Os resultados obtidos indicam que a arquitetura proposta é capaz de lidar com grandes volumes de dados operacionais e de automatizar a identificação de situações potencialmente irregulares. Nesse contexto, o BWCM contribui para o fortalecimento de práticas de fiscalização orientadas por dados, ao reduzir a dependência de análises manuais e permitir uma atuação mais ágil e direcionada por parte da administração tributária. Do ponto de vista arquitetural, o sistema também apresenta características importantes de extensibilidade e modularidade, permitindo a evolução gradual das regras de detecção e a adaptação a novos cenários de fiscalização.

Além de seus aspectos técnicos, o sistema apresenta impacto operacional relevante. A triagem automatizada de grandes volumes de dados fiscais possibilita melhor alocação de recursos de auditoria, reduz o tempo de resposta diante de operações suspeitas e amplia a capacidade analítica das instituições públicas. Dessa forma, o BWCM contribui para iniciativas de transformação digital no setor público, alinhando-se a abordagens contemporâneas de governo orientado por dados.

Apesar dos avanços alcançados, o trabalho apresenta algumas limitações. A ausência de um mecanismo formal de versionamento das regras pode dificultar a rastreabilidade histórica das decisões do sistema. Além disso, a lógica de detecção atualmente é baseada em regras determinísticas, o que limita a capacidade de identificar padrões complexos ou comportamentos anômalos não previamente especificados. Outro aspecto que pode ser aprimorado é a explicabilidade das notificações geradas, permitindo descrições mais detalhadas das condições que originaram cada alerta.

Como trabalhos futuros, destacam-se algumas direções de evolução para o sistema. Entre elas, a criação de um ambiente de simulação off-line para avaliação de novas regras fiscais em dados históricos, reduzindo riscos de implantação em produção; a incorporação de novas fontes de dados relevantes para a análise fiscal, como informações logísticas ou financeiras; e a investigação do uso de técnicas de aprendizado de máquina para complementar as regras determinísticas na detecção de padrões de risco. Essas iniciativas podem ampliar significativamente a capacidade analítica do sistema e contribuir para o desenvolvimento de plataformas de fiscalização tributária cada vez mais inteligentes e orientadas por dados.

Referências

Kyrama, S.; Gounaris, A. **Complex Event Processing: Current Status and Considerations for Edge Deployment.**Future Generation Computer Systems, 2025.

Luckham, D. C. **The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.** Addison-Wesley, 2002.

Morabito, V. **Big Data and Analytics for Government Innovation.** Springer, 2015.

Pedrosa, G.; Canedo, E. **Digital Public Service Evaluation in Brazil.** In: *Proceedings of the Latin American Symposium on Digital Government (LASDiGov)*, 2025.

Silva, E. S.; Aguiar, A. C.; Aguiar, Y. P. C.; Oliveira, M. I. **Práticas de Garantia de Qualidade aplicadas ao CACTO.**In: *Proceedings of the Latin American Symposium on Digital Government (LASDiGov)*, 2025.

Vasilopoulou, C.; Theodorakopoulos, L.; Giotopoulos, K. **Big Data Analytics as a Catalyst for Digital Transformation in e-Government.** Government Information Quarterly, 2023.