DM

Deteção de Fraudes num Sistema de Bilhética Eletrónica

DISSERTAÇÃO DE MESTRADO

Raul Norberto Coelho Gonçalves
MESTRADO EM ENGENHARIA INFORMÁTICA

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

janeiro | 2025

Deteção de Fraudes num Sistema de Bilhética Eletrónica

DISSERTAÇÃO DE MESTRADO

Raul Norberto Coelho Gonçalves
MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO
Leonel Domingos Telo Nóbrega

Deteção de fraudes num sistema de bilhética eletrónica

Raul Gonçalves

Universidade da Madeira

Resumo

A fraude em sistemas de bilhética eletrónica representa um desafio crescente para os transportes públicos, originando perdas financeiras significativas e comprometendo a confiança dos passageiros. Esta dissertação explora esse problema no contexto real da Horários do Funchal, propondo uma abordagem de deteção de fraude baseada em regras. A solução desenvolvida recorre a ferramentas *open source* — nomeadamente Pentaho e MySQL — integradas num processo automatizado de ETL, capaz de tratar grandes volumes de dados e identificar comportamentos suspeitos.

Foram definidas e implementadas 13 regras de deteção, testadas rigorosamente com dados manipulados e cenários simulados. Os testes demonstraram a capacidade do sistema em identificar padrões de utilização indevida, classificando os registos como legítimos, inválidos ou fraudulentos, com um elevado grau de precisão. Os resultados mostram que esta abordagem é eficaz, flexível e replicável para outras operadoras de transporte.

Este trabalho não só contribui com uma solução prática e acessível para um problema real, como também estabelece bases sólidas para investigações futuras com técnicas mais avançadas, como *machine learning*.

Palavras-chave: Bilhética eletrónica, Deteção e prevenção de fraudes, Fraude de bilhetes, Sistemas de acesso, Transportes públicos, Segurança em bilhetes.

Abstract

Fraud in electronic ticketing systems poses a growing challenge for public transport, resulting in significant financial losses and undermining passenger trust. This dissertation explores this problem in the real context of Horários do Funchal, proposing a rule-based fraud detection approach. The developed solution uses open source tools — namely Pentaho and MySQL — integrated into an automated ETL process, capable of handling large volumes of data and identifying suspicious behaviors.

Thirteen detection rules were defined and implemented, rigorously tested with manipulated data and simulated scenarios. The tests demonstrated the system's ability to identify patterns of misuse, classifying records as legitimate, invalid, or fraudulent, with a high degree of accuracy. The results show that this approach is effective, flexible, and replicable for other transport operators.

This work not only provides a practical and accessible solution to a real problem, but also establishes a solid foundation for future investigations using more advanced techniques, such as machine learning.

Keywords: Electronic ticketing, Fraud detection and prevention, Ticket fraud, Access systems, Public transport, Ticket security.

Agradecimentos

Esta jornada não teria sido possível sem o apoio, a orientação e o incentivo de várias pessoas que, de forma direta ou indireta, contribuíram significativamente para a concretização deste trabalho.

Em primeiro lugar, expresso a minha mais profunda gratidão ao meu orientador. A sua orientação, a sua paciência e a sua disponibilidade constante foram determinantes em todas as fases deste trabalho. A sua capacidade de guiar, esclarecer e encorajar teve um impacto determinante no seu desenvolvimento.

Um agradecimento muito especial ao Eng. Marco Louro, chefe do departamento tecnológico dos Horários do Funchal, pela confiança que depositou em mim ao apresentar este tópico e me permitir desenvolver este projeto em colaboração com a empresa. A sua abertura e prontidão para ajudar e esclarecer foram essenciais para a concretização deste projeto.

À minha família, não encontro palavras suficientes para expressar o quanto sou grato. Foram o meu alicerce em todos os momentos, nos dias de entusiasmo e nas noites de incerteza. A vossa presença constante e o vosso apoio deram-me força para continuar, mesmo quando o caminho se tornava difícil. Este trabalho é também vosso, pois sem a vossa confiança, incentivo e carinho, esta etapa não teria sido possível.

A todos os amigos, colegas e demais pessoas que, de alguma forma, contribuíram com palavras de encorajamento, sugestões ou simplesmente acreditando em mim.

A todos, o meu mais sincero obrigado.

Índice

Índice de Ilustrações

Imagens

Gráficos

Tabelas

Código

1. Introdução

Este capítulo estabelece o ponto de partida do trabalho, contextualizando o problema da fraude em bilhética eletrónica nos transportes públicos. Apresenta o problema, os objetivos do estudo, a relevância prática da investigação e a estrutura geral da dissertação. Serve de base para enquadrar o tema e preparar os capítulos seguintes.

1.1. Contexto

O crescimento da digitalização nos transportes públicos trouxe avanços significativos na gestão e controlo de bilhética. No entanto, também abriu portas a novos desafios, nomeadamente a ocorrência de fraudes associadas ao uso indevido de bilhetes eletrónicos. Casos como reutilização de bilhetes, falsificações digitais e evasões intencionais têm vindo a causar perdas financeiras substanciais às empresas de transporte, comprometendo a sustentabilidade dos serviços e a equidade no acesso.

Neste contexto, torna-se fundamental desenvolver mecanismos eficazes e automatizados de deteção de fraude, capazes de operar em grandes volumes de dados, identificando padrões anómalos com precisão e em tempo útil.

1.2. Problema

Apesar dos avanços tecnológicos nos sistemas de bilhética, a deteção de fraude continua a ser um problema complexo, dada a diversidade de comportamentos ilícitos e a dificuldade em distingui-los de erros ou variações legítimas.

O presente trabalho foca-se no caso da empresa Horários do Funchal, analisando dados históricos de validações de bilhetes e propondo uma solução concreta para a identificação automática de comportamentos fraudulentos.

O objetivo deste trabalho é desenvolver um sistema de deteção de fraudes baseado em regras, que permita analisar e classificar validações de bilhetes eletrónicos, identificando padrões de comportamento suspeitos.

Os objetivos deste trabalho são:

- Identificar os principais tipos de fraude em bilhética eletrónica e os desafios associados à sua deteção;
- Estudar as abordagens existentes para deteção de fraudes e justificar a escolha de um sistema baseado em regras;
- Desenvolver um sistema de deteção de fraudes recorrendo a ferramentas *open source* (Pentaho e MySQL);
- Testar e validar a eficácia de um conjunto de regras aplicadas a dados reais e manipulados, simulando diferentes cenários de fraude;
- Avaliar a viabilidade da solução em contexto prático, propondo direções futuras de melhoria.

1.3. Contribuições

O trabalho contribui para a área científica ao propor um sistema de deteção de fraudes baseado em regras no contexto dos bilhetes eletrónicos em transportes públicos. Ele apresenta uma abordagem detalhada para a análise de

dados e validação de registos históricos, destacando como a aplicação de regras pode identificar inconsistências e comportamentos fraudulentos. A pesquisa também explora o uso de ferramentas como o Pentaho e o MySQL, demonstrando a sua capacidade em manipular e analisar grandes volumes de dados.

Do ponto de vista das boas práticas da área de engenharia informática, o trabalho destaca-se pelos seguintes pontos principais:

- ETL Automatizado: A implementação de um processo de ETL (Extração, Transformação e Carregamento) automatizado facilita o tratamento de dados ao eliminar informações irrelevantes e gerir dados ausente de forma eficiente. Além disso, possibilita uma manipulação mais otimizada de grandes volumes de dados e a classificação das validações. Como resultado, reduz a ocorrência de erros manuais e melhora a eficiência de todo o processo.
- Uso de ferramentas *Open Source*: A escolha pelo Pentaho e o MySQL, ferramentas de código aberto, promove flexibilidade, escalabilidade e acessibilidade ao projeto.
- Testes rigorosos: As regras desenvolvidas foram testadas individualmente com dados clonados e modificados para assegurar a precisão e confiabilidade das classificações.
- Melhoria contínua: O trabalho estabelece uma base de desenvolvimento futuro e menciona a importância da exploração contínua de novas técnicas, como o *machine learning*, visando aumentar a robustez e a eficiência do sistema.

1.4. Estrutura da Dissertação

O presente trabalho está estruturado em quatro capítulos principais:

No primeiro capítulo, será apresentada uma revisão bibliográfica, onde aborda o conceito de fraudes, os diversos tipos e formas de fraude de bilhetes, as suas consequências, e uma análise de soluções e métodos de deteção que podem ser aplicados. Este capítulo serve como um alicerce teórico que fornece uma base para a compreensão das complexidades inerentes ao problema em questão.

O segundo capítulo irá definir e analisar o problema apresentado pela Horários do Funchal, de forma a compreender detalhadamente os requisitos da empresa em relação à deteção de fraudes de bilhetes eletrónicos. Em seguida, será apresentado o sistema de deteção a ser desenvolvido e implementado, bem como a exploração de ferramentas existentes e a seleção daquela que será utilizada. Finalmente, será discutido que tipo de método é que será utilizado para avaliar a solução proposta. Este capítulo tem como objetivo oferecer uma compreensão mais profunda e esclarecedora da proposta adotada para abordar e solucionar o problema definido no início deste capítulo.

No terceiro capítulo, serão abordadas considerações sobre a viabilidade da proposta apresentada, proporcionando uma análise das condições circundantes ao projeto. Isto avaliará a facilidade ou complexidade da realização do projeto, levando em conta fatores que irão influenciar diretamente a sua concretização.

No quarto e último capítulo, abordaremos em detalhe o desenvolvimento do sistema de deteção de fraude. Inicialmente, faremos uma breve recapitulação da ferramenta Pentaho Data Integration (Spoon), destacando as

vantagens que ela agrega ao projeto. Em seguida, explicaremos a escolha do MySQL como sistema de gestão de base de dados, detalhando o seu papel no armazenamento e manipulação das informações essenciais, bem como uma descrição aprofundada dos campos da tabela, que constituem uma das principais estruturas de suporte do sistema. Prosseguiremos com uma análise individual das transformações aplicadas, discutindo os seus objetivos e a função de cada componente envolvido. Em sequência, apresentaremos as principais regras de deteção definidas, esclarecendo os seus propósitos. Cada regra será testada individualmente, e os resultados obtidos serão analisados e discutidos, com igual destaque para as observações e dificuldades encontradas ao longo do processo. Por fim, descreveremos a metodologia aplicada para o teste final, incluindo uma análise estatística dos resultados, de modo a assegurar a robustez do sistema de deteção de fraudes baseado em regras.

Este trabalho encerra com uma conclusão que sintetiza os pontos mais importantes que serão abordados ao longo dos quatro capítulos, destacando, igualmente, os desafios enfrentados.

2. Revisão bibliográfica

Este capítulo apresenta uma revisão bibliográfica focada no fenómeno da fraude, com ênfase nas suas manifestações no setor dos transportes públicos e, em particular, na bilhética eletrónica. O objetivo principal é fornecer uma base teórica sólida que sustente a proposta de deteção de fraude desenvolvida nesta dissertação.

Para isso, são explorados conceitos fundamentais de fraude, diferentes classificações e formas de atuação, causas subjacentes, consequências para as organizações e indivíduos, bem como estratégias e tecnologias utilizadas na sua prevenção e deteção.

A inclusão dos conceitos gerais de fraude justifica-se por três razões principais: (1) proporciona um enquadramento académico mais completo; (2) ajuda a definir com precisão o que constitui fraude, distinguindo-a de erros operacionais — o que é essencial para a construção das regras de deteção; e (3) assegura que a metodologia adotada tem potencial de aplicação a outros contextos, tornando-a mais robusta e transferível.

Esta análise teórica permite compreender a complexidade do problema e identificar lacunas e oportunidades que justificam a abordagem adotada nos capítulos seguintes.

2.1. Conceito de fraude

A fraude é um ato ilegal que visa enganar ou retirar o acesso a algo que pertence à vítima por direito, com o objetivo de fornecer ao autor um benefício ilegal ou injusto. O autor que desempenha esta atividade fraudulenta pode assumir diversas entidades como, uma pessoa, um grupo de pessoas ou uma empresa. A fraude pode ainda assumir vários tipos, por exemplo: a fraude fiscal, fraude com cartões de crédito, fraudes eletrónicas, fraude na segurança, fraude por falência, entre outros [1]. As áreas que apresentam benefícios mais altos, tal como a financeira, são mais propensas a fraudes [2]. Neste capítulo, serão ainda abordadas outras formas de fraudes, cada uma com uma explicação sucinta.

Na realidade, não existe uma definição ou uma infração específica para a fraude, isto porque ela aplica-se em diversas áreas e pode assumir vários tipos, formas e processos, dependendo dos seus objetivos, de enganar ou prejudicar uma vítima. Isso acontece porque o objetivo da fraude é obter o que se pretende e não o processo que é adotado. Como foi referido anteriormente, apesar de não haver uma infração específica para a fraude, a lei contém um conceito de fraude altamente abrangente com o objetivo de detetar o maior número de casos possíveis de fraude, de forma a punir os indivíduos que desempenham o papel de fraudulentos [3].

De uma forma simples, uma fraude acontece quando todas as seguintes etapas são cumpridas:

1. O autor cria uma representação falsa, de forma intencional, de um fato ou evento importante;
2. A vítima aceita a representação;
3. A vítima confia e age com base na representação;
4. A vítima perde o seu dinheiro e/ou as suas propriedades [4].

2.2. Distinção entre fraude ocupacional e fraude organizacional

O conceito de fraude pode ser dividido em duas partes: Fraude ocupacional e fraude organizacional. Estamos perante uma fraude ocupacional quando um funcionário, gerente ou proprietário de uma empresa comete atos fraudulentos com o objetivo de deteriorar a própria empresa. Esta forma de fraude normalmente é bem estruturada, planeada e é capaz de apresentar-se em diversas formas, por exemplo, a corrupção, roubo de recursos, declarações fraudulentas, entre outros. Já a fraude organizacional é desempenhada por funcionários de uma empresa, que agem de forma a beneficiar a empresa como um todo e não individualmente. Recapitulando, enquanto a fraude ocupacional beneficia apenas o autor da fraude, por outro lado, fraude organizacional beneficia a empresa que realiza o ato fraudulento [5].

2.3. Os elementos constitutivos da fraude

Os elementos essenciais para a configuração de uma fraude são:

1. **Motivo** - Existe sempre um motivo pela qual a fraude está associada. O endividamento e o desespero por dinheiro podem ser um motivo para alguém defraudar a vítima.
2. **Atração** - Os autores quando praticam o ato da fraude tencionam sempre ganhar ou beneficiar-se de algo. Esta atração pode ser motivada pelo ganho de dinheiro, bens, ou esconder algo.
3. **Oportunidade** - Não é possível haver fraude se não houver oportunidades. A oportunidade é uma chance ou uma brecha que o fraudulento tem para executar a fraude. Normalmente, estas oportunidades ocorrem quando há uma fragilidade ou um uso indevido das medidas que permitem defender e detetar as fraudes.
4. **Meio** - Consiste nas habilidades e técnicas que o atacante adota para executar a fraude. Portanto, é importante que o atacante possua capacidades de detetar as oportunidades, para então definir os seus meios [6].

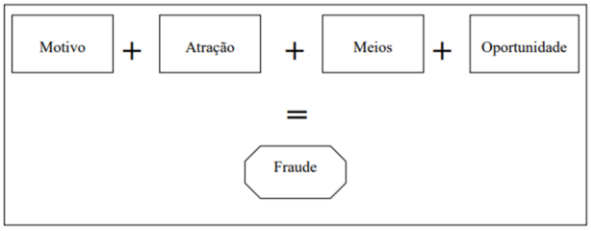


Imagem 1 - Componentes da fraude [6]

A imagem 2 apresenta um caso ilustrativo da implementação dos elementos constitutivos da fraude.

Motivo	Pagando altas quantias de pensão
Atração	Dinheiro, Mercadorias.
Meios	Habilidades Técnicas e outras presentes
Oportunidade	Falta de Controle Interno, possibilidade de Encobrimento.

Imagem 2 - Exemplo simples dos elementos essenciais para a configuração da fraude [6]

2.4. Os desafios da deteção e visualização de fraudes: exemplo e motivos

Muitas das vezes a deteção e visualização de fraudes é realmente desafiadora por diversos motivos. Por exemplo, imaginemos o cenário onde uma caixa de fósforos anuncia 100 unidades, mas num certo dia, passam a vir com 95, mas continua a anunciar a mesma quantidade, muito dificilmente as pessoas iriam notar a falta de fósforos. Os motivos são os seguintes:

- Novos tipos e formas de fraude erguem-se todos os dias que se passam de despercebidas e que são visualizadas e detetadas passado algum tempo desde o início do ato;
- Existe uma dificuldade em distinguir as ações normais e as ações fraudulentas, mesmo que ambas sejam intencionais, devido às normas e regras que são criadas [5]. Isso se justifica pelo facto de que a fraude pode assumir vários tipos e formas.
- Por vezes as fraudes podem assumir formas tão simples, facilmente executáveis e não detetáveis, que têm como foco o desconhecimento e o desejo da angariação de dinheiro de forma fácil das pessoas. Por outro lado, as fraudes também podem assumir formas muito complexas, que acabam por ser difíceis de detetar e compreender. Neste caso, a fraude é desempenhada profissionalmente por pessoas especialistas nesta área.
- Em alguns casos de fraude, os prejudicados não se reconhecem como vítimas, ou então existe uma dificuldade maior em reconhecer quem são elas [5].

2.5. Alguns tipos e formas de fraude

Conforme mencionado repetidamente neste trabalho, a fraude apresenta-se sob diversos tipos e formas. Alguns exemplos de tipos de fraudes são os seguintes:

- **Furto** - O furto é quando alguém tira algo que não é seu e nunca devolve. Um exemplo é quando alguém tira dinheiro de uma caixa registadora que não é sua.
- **Pirataria** - A pirataria é quando alguém finge que inventou ou criou algo, quando na realidade trata-se de uma cópia.
- **Falsificação** - A falsificação é quando alguém possui algo falso, que tem intenções de o vender ou fazer as pessoas acreditarem que é verdadeiro.
- **Espionagem** - A espionagem dá-se quando um funcionário de uma empresa fornece informações confidenciais para a concorrência em troca de dinheiro ou outros benefícios. Outro caso de espionagem é quando um funcionário possui informações sensíveis de uma empresa, com o objetivo de prejudicar a mesma através de denúncias [6].
- **Perjúrio** - É quando alguém mente ou faz declarações falsas durante um juramento, em que a pessoa é obrigada a dizer a verdade [6], [7].

- **Vendas pela Internet** - A fraude é cada vez mais comum nesta área, pois cada vez mais as pessoas compram artigos pela internet. Este tipo de fraude envolve a falsificação, ou então um cliente paga por um produto ou serviço, mas nunca é enviado ou entregue.
- **Desvio da direção do site** - É um tipo de fraude em que os atacantes criam imitações quase perfeitas de sites confiáveis, Amazon, eBay, Paypal, para induzir as vítimas, onde redireciona os consumidores para outros sites maliciosos para inserirem os dados do cartão de crédito, bancários, PayPal, entre outros.
- **Fraudes de trabalho em casa** - É quando os fraudulentos prometem um salário para as pessoas que se inscrevem no seu "trabalho em casa", exigindo antecipadamente um pagamento, convencendo as vítimas com a promessa de obter grandes quantidades de lucro num curto espaço de tempo. Um exemplo deste tipo de fraude é o caso da Telexfree.
- **Fraude com cartão de crédito** - Com o aparecimento do dinheiro eletrónico, este tipo de fraude tornou-se mais banal. Após obter as informações do cartão de crédito de alguém, através de vários métodos, onde já foram referidos alguns, o autor da fraude tem ao seu dispor o capital da vítima e com isso, poderá realizar compras sem a autorização e o conhecimento do titular.
- **Eliminação de dívidas** - Muitas pessoas têm dívidas numerosas ou enormes e, por vezes, são incapazes de liquidá-las. Esses indivíduos acabam entrando em desespero, buscando dinheiro para pagar as dívidas, pois, caso contrário, irão perder os seus bens. Os fraudadores aproveitam-se e oferecem às vítimas oportunidades para eliminarem as suas dívidas, cobrando um pagamento antecipado. A vítima, desesperada, paga esse valor e fornece informações sensíveis, como dados do cartão de crédito ou do banco, e não recebe nada em troca. O fraudador, então, pode retirar todo o capital da vítima e até vender essas informações para outros criminosos [2].

2.6. Um exemplo verídico: o caso da Enron

Para ilustrar a gravidade da fraude, vejamos um exemplo das consequências de uma fraude num caso real:

Em 2001, a Enron, que é uma empresa sediada nos Estados Unidos da América, foi alvo de um enorme golpe de fraude ocupacional. Os executivos da empresa utilizaram diversas técnicas com objetivo de disfarçar as condições financeiras da mesma, como mentir a quantidade de receita e de lucro. Quando esta fraude foi detetada e descoberta, as ações da empresa diminuíram de 90$ para menos de 1$ em cerca de um ano. Com isso, os funcionários acabaram por perder as suas ações e os seus empregos quando a Enron faliu [1].

2.7. Prevenir a fraude: dicas e recomendações

Para concluir, a fraude é um problema real e sério que afeta milhares de indivíduos e organizações. Para se protegerem e combaterem a fraude, é importante seguirem algumas boas práticas, que podem ser simples, mas muito eficazes. Neste ponto, serão apresentadas algumas boas práticas mais focadas para as pessoas e outras que se aplicam melhor nas empresas.

Uma boa prática essencial, e talvez a mais importante, é nunca fornecer informações pessoais a desconhecidos, sejam eles indivíduos ou organizações, por qualquer meio de comunicação. As pessoas devem ter sempre em conta que uma empresa ou banco legítimo jamais solicita nenhum pagamento antecipado por qualquer motivo. Se receberem tal pedido, devem investigar e consultar as autoridades sobre a legitimidade da empresa antes de efetuar o pagamento. Quando alguém pretende comprar um serviço ou produto, deve exigir ao vendedor o pagamento em dinheiro vivo, de forma a se proteger da fraude que usa os dados da transferência bancária, por exemplo [2].

No caso empresarial, é extremamente importante que as entidades estejam conscientes que a organização se encontra exposta e adotem os métodos de gestão mais apropriados, para garantirem um nível de segurança mínimo. A avaliação de riscos é um aspeto essencial para uma empresa, por isso, existem algumas etapas que devem ser seguidas durante a avaliação de riscos de uma fraude:

1. Identificar as vulnerabilidades que podem exploradas pela fraude;
2. Reconhecer os tipos de fraude existentes e aprender a detetá-los, preveni-los e combatê-los na empresa;
3. Desenvolver um plano com medidas de prevenção de fraudes ajustado à empresa e identificar as suas falhas;
4. Avaliar o grau de eficácia das medidas de prevenção de fraudes criadas;
5. Provar e documentar a avaliação do risco de fraude.

Desta forma, as empresas devem se proteger de forma eficaz, e para isso, é fundamental que conheçam as suas vulnerabilidades em relação às fraudes, as quais dependem de como realizam a avaliação, e implementem medidas de mitigação de fraudes eficientes. Adicionalmente, deve haver uma especial atenção às motivações que originam a fraude, de modo que seja possível detetá-las, antes que a fraude se concretize [8].

2.8. Fraudes de bilhetes

2.8.1. A fraude de bilhetes nos transportes públicos: origem e tipos de fraude

Os transportes públicos são um meio essencial para o deslocamento das pessoas, uma vez que a grande maioria delas dependem desses serviços no seu dia a dia. Esses transportes têm dado e continuam a dar um contributo enorme para a sociedade, tanto no aspeto econômico quanto ambiental.

Porém, esses serviços têm um custo e as empresas de transportes públicos dependem da receita gerada pelas vendas de bilhetes para se sustentarem, oferecerem melhores condições aos passageiros através da aquisição de transportes novos e realizarem as manutenções dos veículos, entre outros [9].

Esse custo refere-se à aquisição de bilhetes por parte dos utilizadores do serviço, que devem mostrá-los aos inspetores como prova de que estão a viajar de forma legal. Mas, sempre que há uma transação monetária envolvida, haverá sempre alguém que tentará cometer fraudes [10].

Os bilhetes antigos eram muito simples e básicos, como se pode ver na Imagem 3. O seu controlo e validação era apenas através da inspeção visual, o que podia demorar muito tempo, gerando filas e transtornos, e era propício a erros

humanos. Adicionalmente, era muito fácil falsificar bilhetes, que por vezes, eram quase idênticos aos originais, dificultando ou impossibilitando a sua deteção apenas visualmente [10].

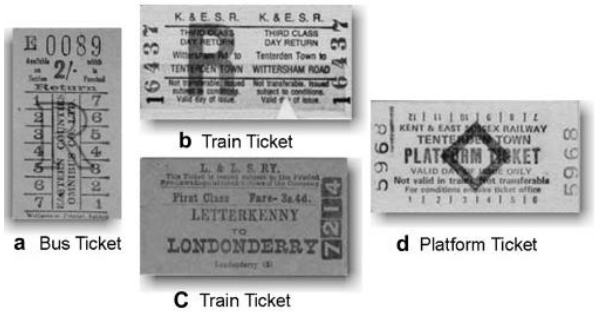


Imagem 3 - Bilhetes antigos [10]

As fraudes que se destacam neste tema são:

- A **evasão de tarifa** ou **fraude de tarifa**, também chamada de **viagem gratuita**, é o ato de usar transportes públicos sem pagar o valor exigido para a viagem. Esta prática é intencional e envolve não comprar ou não validar o bilhete.
- A **fraude de bilhetes** ou **falsificação de bilhetes** é o ato de produzir ilegalmente bilhetes falsos, que são usados para ludibriar o sistema de controlo e serem aceites como válidos.

Estas fraudes causam grandes danos às empresas de transportes públicos, que resultam numa quebra de receita. Por isso, é importante implementar novos métodos de prevenção, deteção e fiscalização mais modernos e eficientes, uma vez que o aparecimento de novas tecnologias permitem que as fraudes se tornem mais variadas e complexas [9].

2.8.2.Evasão de tarifa: Conceito, desafios e medidas de defesa

É essencial que as empresas de transportes públicos não se foquem apenas nos casos de evasão de tarifa intencional, em que um passageiro clandestino usa o serviço ilegalmente sem pagar o bilhete, propositadamente, para obter benefício próprio ou por prazer de enganar as pessoas ou as organizações, ou pela adrenalina que sentem de não serem apanhados. Mas também devem prevenir os casos de evasão de tarifa involuntária, que podem ocorrer quando um passageiro se esquece de comprar ou validar o bilhete, de verificar o período ou área de validade ou não compreende o sistema de venda, compra o bilhete errado e usa-o sem querer. Portanto, é importante que as empresas de transportes públicos estejam atentas a todas as situações e que reconheçam que nem todos os casos são propositados.

Segundo um caso de estudo nesta referência, os especialistas defendem que controlos mais rigorosos e intensos podem ajudar a reduzir as fraudes por evasão de tarifas. Eles sugerem que o número de inspetores que validam os bilhetes seja maior, de forma a desencorajar as pessoas que usam ilegalmente os transportes públicos. Outra medida

eficaz é exigir que as pessoas entrem sempre pela porta da frente, obrigando-as a passarem pelo motorista, que por sua vez vende e valida os bilhetes. A implementação de um sistema de bilhética eletrónica também pode ajudar a evitar a evasão de tarifa [9].

2.8.3. Fraude de bilhetes: Conceito, desafios e medidas de defesa

O problema da fraude de bilhetes é complexo e sensível para as empresas de transportes públicos, pois muitos desses casos são difíceis de detetar, como veremos a seguir. Segundo a definição que apresentamos anteriormente, este tipo de fraude envolve não só a evasão ao pagamento de um bilhete, mas também um maior esforço monetário e temporal por parte dos criminosos. Estes, por vezes, investem tanto neste tipo de fraude, que chegam a desenvolver ou comprar equipamentos especializados que produzem bilhetes falsificados com um grau de detalhe impressionante, tornando muito difícil a distinção entre um bilhete verdadeiro de um falso, num curto espaço de tempo, sem equipamentos eletrónicos avançados. É por isso que este é um tema muito sensível para estas empresas. Este é um assunto tão sério, que há rumores de que alguns representantes de empresas de transportes públicos que só conseguiram detetar alguns casos porque havia erros de escrita no bilhete ou porque os números de série do bilhete eram sempre os mesmos.

A fraude de bilhetes pode assumir várias formas, umas mais graves do que outras. Por exemplo, um bilhete comprado de forma legítima, pode ser usado duas vezes se o carimbo na primeira viagem for muito leve, de modo que a sua marca não seja legível, e assim, na segunda viagem, o mesmo bilhete é marcado com o carimbo novamente, cobrindo totalmente o primeiro. Outro exemplo, mais grave, é quando um criminoso consegue aceder às máquinas de impressão de bilhetes da empresa de transportes públicos e rouba os rolos de impressão. Desta forma, o criminoso pode facilmente produzir bilhetes idênticos aos originais para uso pessoal ou para venda.

As empresas investem muito em tecnologias para combater a fraude de bilhetes, como adicionar elementos de segurança a eles, por exemplo hologramas, códigos de barras, números de série e papel especial, mas esses elementos podem ser comprometidos após algum tempo. Isto implica que eles precisam de ser trocados com frequência. Os especialistas afirmam que esses elementos são importantes e ajudam, mas não resolvem o problema, pois essa mudança constante não compensa. Por isso, os bilhetes eletrónicos são uma opção, pois possibilitam a alteração desses elementos de forma simples [9].

2.8.4. Alguns motivos por trás da fraude nos transportes públicos

A mesma investigação ainda analisou os motivos que levam as pessoas a cometerem fraudes em transportes públicos. Os principais fatores são: a grande fragilidade do sistema de controlo de bilhetes e sanções muito pouco severas. Além disso, as pessoas cometem fraudes pela insatisfação com a relação de preço e a qualidade do serviço prestado, ou simplesmente pela distância da viagem. No entanto, nem todas as fraudes são intencionais e é preciso considerar essa possibilidade, como foi argumentado anteriormente [9].

2.8.5. Combate às fraudes nos transportes públicos: prejuízos e soluções

Com isto, é fundamental que as empresas de transportes públicos não subestimem as ocorrências de fraudes, sejam elas graves ou não. Pois uma fraude simples pode causar prejuízos financeiros menores, mas frequentes, por ser fácil de praticar, enquanto uma fraude grave pode causar um grande prejuízo de uma só vez. Além do dano financeiro, as fraudes também afetam a dignidade, o respeito e a honestidade das pessoas que pagam pelas viagens de forma legal. Para prevenir esses tipos de fraude, as empresas de transportes públicos devem implementar métodos para combatê-las, levando em conta a natureza do problema, pois como vimos, a fraude pode ter várias formas. Outro fator que influencia nessa decisão é o tipo de sistema de acesso utilizado e, durante a implementação desses métodos, é preciso avaliar se vale a pena manter ou alterar o sistema de acesso. Os tipos de sistemas de acesso serão discutidos a seguir [9].

Como foi dito anteriormente, as empresas de transportes públicos podem adotar dois tipos de sistemas de acesso que ajudam a combater as fraudes: Sistemas de acesso aberto e sistemas de acesso fechado. Os sistemas de acesso aberto não possuem quaisquer tipos de controlo de bilhética à entrada de um lugar ou de um transporte público. Esse controlo é assegurado por inspetores, fardados ou não, que verificam se o bilhete é válido ou não, após o passageiro estar a usufruir do serviço [9]. Este tipo de sistema é mais usado em locais onde há um grande fluxo de passageiros, onde não vale a pena recrutar uma grande quantidade de inspetores para controlarem os bilhetes [10]. Por outro lado, os sistemas de acesso fechado já possuem controlos de bilhetes à entrada, que normalmente é realizada por pessoas, como os motoristas dos autocarros, que vendem e validam os bilhetes [9].

Outra medida que pode ser eficaz é aumentar as sanções em caso de fraude. Por outro lado, isso também pode prejudicar as pessoas honestas que enfrentaram dificuldades em adquirir os bilhetes corretos, por exemplo. Por isso, é que deve ser considerado todos os casos, como já mencionamos anteriormente [10].

Para demonstrar que estes sistemas são uma boa forma das empresas de transportes públicos protegerem-se das fraudes, um caso de estudo, liderado por Thomas Muller, verificou que houve uma diminuição de alguns tipos de fraude, como os passageiros sem bilhete, em Berlim, após um ano a sua implementação.

O *feedback* desta implementação foi positivo, não foram registados problemas operacionais e, apesar dos controlos, que normalmente exigem tempo, as viagens não ficaram mais demoradas. Mas o mais importante de tudo é que essa implementação foi bem aceite pelas pessoas [9].

2.8.6. Resumo das estratégias para prevenir as fraudes

Com base no que foi escrito anteriormente e no caso de estudo, que também enumera algumas medidas de prevenção, podemos enumerar algumas medidas que ajudam a evitar as fraudes nos transportes públicos. Algumas delas são:

- Aumentar o número de inspetores de modo a intensificar o controlo de bilhetes;
- Elevar o valor das multas para as pessoas que cometem o crime de forma intencional;

- Imprimir os bilhetes com elementos de segurança, como hologramas, códigos de barras, números de série e papel especial;
- Adotar um sistema de bilhética eletrónica;
- Instalar barreiras;
- Exigir o acesso pela porta frontal do autocarro e validação do bilhete junto ao motorista;
- Melhorar a qualidade de serviço;
- Praticar preços justos e que são compreensíveis pelas pessoas.

Estas são as medidas mais importantes, mas há outras que podem ser consideradas. É essencial que as empresas de transportes públicos avaliem os métodos e que escolham os mais adequados, pois a combinação de várias técnicas de prevenção é a melhor solução para combater as fraudes [9].

2.8.7. Análise comparativa de alguns sistemas de controlo de bilhetes: Prós e contras

Após apresentar algumas medidas, podemos agora indicar, de forma muito resumida, as vantagens e desvantagens de alguns sistemas de controlo:

- As barreiras mecânicas são uma medida eficaz de controlo, pois dificilmente há erros humanos. Por outro lado, elas exigem um alto investimento, ocupam muito espaço, podem gerar congestionamentos de pessoas em horas de pico, dificultam a saída em caso de emergência e exigem uma manutenção e reparações constantes, pois são sistemas mecânicos.
- Os sistemas eletrónicos são rápidos e dispensam o contacto físico. Desta forma, a falsificação de bilhetes torna-se numa tarefa difícil, o que contribui para uma diminuição dos casos de fraude. Eles ainda permitem recolher dados que podem ser usados para *marketing*, como promoções e ofertas. No entanto, eles exigem um alto custo de implementação, são vulneráveis a falhas e podem ocorrer perdas de dados.
- A fiscalização por *staff* é de longe o método mais comum. Permite organizar e gerir os controlos de forma flexível, adaptando-se às diferentes situações. No entanto, este exige um custo elevado para manter o *staff*, pode gerar conflitos por parte dos passageiros e está sujeito a erros humanos, como não detetar um bilhete falso [9].

2.8.8. Reflexão

Concluindo, as fraudes de bilhetes são uma ameaça grave e frequente, que pode causar grandes prejuízos às empresas de transportes públicos. Portanto, é essencial que as empresas se preocupem com este problema, independentemente da sua complexidade, e não as subestimem. Foram ainda enumeradas algumas medidas simples, mas eficazes, que podem contribuir para a prevenção contra fraudes, tais como aumentar o número de inspetores, melhorar a qualidade do serviço, disponibilizar formas de aquisição de bilhetes claras e acessíveis, entre outras. A evolução das tecnologias permitiu a implementação de sistemas de acesso abertos e fechados, que vieram garantir

uma maior segurança ao sistema de controlo, pois dificultam o engano dos métodos de verificação. O surgimento dos cartões físicos eletrónicos, que possuem um *chip* e a tecnologia NFC, garantem ainda mais rapidez e segurança, pois é difícil de falsificar, pode ser facilmente anulado / cancelado em caso de suspeita de fraude, o cartão possui um saldo e os bilhetes e os passes podem ser armazenados no cartão. Por fim, é importante que as empresas de transportes públicos busquem as melhores estratégias, analisando sempre as vantagens e desvantagens de cada uma, das quais algumas foram abordadas anteriormente, para ver qual é a mais adequada, e que a combinação de várias medidas de prevenção é muito mais eficaz do que apenas uma [9], [10].

2.9. Fraude de bilhetes eletrónicos e digitais

2.9.1. Bilhetes eletrónicos e digitais: nova forma de compra e venda no transporte público

Desde que foi criado por um matemático em 1662 por Blaise Pascal [11], o transporte público passou por diversas transformações devido à evolução tecnológica e às necessidades da sociedade. Este meio de transporte teve de procurar soluções mais eficientes e eficientes para atender às necessidades de mobilidade urbana ao longo dos anos.

Mais recentemente, uma dessas inovações foi a implementação da bilhética eletrónica [12]. A bilhética eletrónica e digital, são um novo método de venda e compra de bilhetes para os transportes públicos [13].

Os bilhetes eletrónicos possuem uma banda magnética na parte de trás e foram criados para melhorar o controlo destes, pois a sua verificação era mais rápida, automática e eficaz, e, consequentemente, aumenta a taxa de deteção de bilhetes falsos. Com isso, os sistemas de acesso fechados ficaram mais seguros e práticos. No entanto, estes bilhetes ainda podem ser facilmente copiados, mas mesmo assim, representam um avanço em relação aos bilhetes em papel, onde o controlo era feito por furos e visualmente, embora o nível de segurança se mantivesse praticamente o mesmo. Outro problema associado a estes bilhetes eletrónicos é que eles possuem apenas uma autenticação de um fator, ou seja, o próprio bilhete, o que significa que se alguém o encontrar, roubar ou copiar, pode usar a viagem da pessoa que o comprou de forma legal. Poder-se-ia usar um bilhete com uma autenticação de dois fatores, semelhante à dos cartões de crédito, em que o primeiro fator seria o bilhete e o segundo seria o PIN (*Personal Identification Number* ou Número de Identificação Pessoal), mas isso poderia causar lentidão quando as pessoas quisessem entrar na estação ou no autocarro, por exemplo. Apesar de todos os problemas, estes bilhetes eram bastante utilizados, pois o sistema de acesso fechado e a infraestrutura que envolvia estes bilhetes eletrónicos proporcionam uma boa base de segurança e de combate à fraude [10].

Os equipamentos que estão associados aos bilhetes eletrónicos e que fazem o controlo e a validação dos bilhetes são as barreiras físicas ou técnicas. Elas podem ser torniquetes, portas, portões automáticos, etc. Contudo, as empresas não devem prescindir totalmente das inspeções realizadas por pessoas, pois estas barreiras podem ser facilmente contornadas [9].

Os bilhetes eletrónicos posteriormente evoluíram para uns cartões que têm um chip embutido no plástico e que podem comunicar com equipamentos sem contacto físico, ou seja, sem fios. Estes cartões permitem registar e monitorizar as viagens que as pessoas fazem e desativar um determinado cartão em caso de suspeita de fraude.

Adicionalmente, estes cartões têm um saldo, o que significa que as pessoas não necessitam de comprar bilhetes a cada viagem que efetuam, evitando filas e tornando tudo mais eficaz e prático. Os produtos da viagem, como passes, bilhetes, podem ser guardados no cartão, para que a pessoa possa usá-los quando precisar. A funcionalidade de comunicação sem fios para verificar e validar o cartão proporciona transações mais rápidas e seguras. Os bilhetes eletrónicos com banda magnética não tinham esta capacidade, pois eles tinham uma capacidade de armazenamento muito limitada, o que não acontece com estes cartões eletrónicos com chip e comunicação sem fios [10].

Os sistemas que estão associados a estes cartões são os sistemas mecânicos e os sistemas eletrónicos. Nos sistemas mecânicos, a leitura do bilhete é feita por meio de um leitor de cartões, enquanto que nos sistemas eletrónicos, a leitura é feita por meio de equipamentos com a tecnologia RFID (*Radio Frequency Identification* ou Identificação por radiofrequência), neste caso não existe uma conexão física entre o bilhete e o equipamento, pois a leitura e a identificação dos dados para verificar se o bilhete é válido ou não são feitas por ondas de rádio. Com este tipo de tecnologia, a leitura e validação dos bilhetes é muito mais rápida, pois normalmente não requerem nenhum tipo de barreira ou leitores de cartões, que costumam atrasar o processo [9]. Esta tecnologia, embora muito improvável, permite que um atacante com outro leitor RFID possa roubar algo da vítima, mas isso é muito difícil, pois o atacante precisa de estar perto dela [14], porque os sinais RFID têm um alcance local e limitado [12].

Os bilhetes digitais, por sua vez, representam uma nova evolução neste cenário [12]. Este método tem capacidade de se tornar no mais utilizado na área da bilhética, de acordo com várias evidências. Estes bilhetes são completamente virtuais e vendidos eletronicamente, sendo vistos como um registo de uma reserva que fica armazenada em dispositivos eletrónicos, como computadores, tanto da empresa que os comercializa como do cliente que os adquire. Adicionalmente, as informações dos bilhetes digitais ficam todas armazenadas numa base de dados, o que elimina a necessidade de imprimir bilhetes em papel [13]. Estes sistemas eletrónicos e estas informações que são guardadas nas bases de dados permitem fazer um registo dos utilizadores, um maior controlo na venda de bilhetes, carregar créditos e usá-los e emitir relatórios de gestão que permitem acompanhar os dados [12]. A bilhética eletrónica veio proporcionar uma maior rapidez e praticabilidade, pois permite que as pessoas comprem os seus bilhetes em qualquer lugar e a qualquer hora, sem enfrentar filas longas ou esperar pelo bilhete, aumentando assim a satisfação do cliente. A compra destes bilhetes pode ser feita por meio de *call-centers*, pela internet, entre outros meios, e depois podem ser descarregados no *smartphone*, armazenados nos cartões físicos eletrónicos, ou consultados no próprio *website* da empresa, o que significa que eles estão disponíveis e acessíveis a qualquer momento. Para utilizar o transporte público, o passageiro só precisa apresentar o número de confirmação juntamente com o seu cartão de identificação ou, alternativamente, fazer a leitura do bilhete por meio da tecnologia NFC do seu *smartphone* ou pelo código QR. A bilhética eletrónica ainda permite às empresas poupar despesas monetárias com *staff* e reduz a fraude por falsificação, por meio da impressão. No entanto, apesar de todos estas vantagens, os bilhetes digitais apresentam alguns problemas e outras formas de cometer fraudes [13].

2.9.2. Motivações para a fraude de bilhetes digitais

As fraudes de bilhetes digitais é um problema sério que afeta milhares de pessoas e empresas todos os anos e às necessidades da sociedade. Existem diversas motivações que levam os criminosos a se aproveitarem dos bilhetes digitais para cometerem fraudes, tais como:

- A vulnerabilidade dos códigos de barras e dos códigos QR reside na facilidade com que podem ser copiados e replicados. Isso implica que, se alguém utilizar previamente um bilhete falso contendo o mesmo código que o bilhete original, este poderá ser validado e aceite na entrada, enquanto o original, mesmo sendo completamente legítimo, será recusado [15]. Esta fragilidade dos bilhetes digitais torna-os suscetíveis à duplicação e falsificação, permitindo que os fraudadores comercializem bilhetes falsos para múltiplas pessoas, resultando em lucros ilegais significativos [13].
- Os fraudadores aproveitam-se dos serviços em que há uma grande procura e compra de bilhetes. Como esses bilhetes esgotam-se rapidamente, as pessoas ficam desesperadas para conseguir um, mesmo que tenham que pagar mais caro. Desta forma, elas acabam procurando em *websites* não oficiais e clicando em *links* suspeitos, onde os fraudadores acabam por se beneficiar [15], [16].
- Os fraudadores aproveitam-se de contas com credenciais fracas que possuem reservas de bilhetes, às vezes caros, onde usam mecanismos automáticos para descobrir as combinações de nome de utilizador e palavra-chave ou usam técnicas de *phishing* para roubar as credenciais para transferir os bilhetes para si mesmo [15].
- Muitas vezes, os compradores, principalmente os mais inexperientes, são enganados pela especulação, ou seja, os fraudadores criam anúncios de venda de bilhetes digitais que eles não possuem, mas que podem obter e vender por um preço muito mais alto do que o original, lucrando com isso [15].

2.9.3. Fraudes: Métodos e Exemplos

Diariamente, ocorrem incidentes que envolvem a utilização fraudulenta de bilhetes digitais adquiridos de maneira ilegal, seja para uso pessoal, cúmplices ou revenda. Uma das formas mais usadas de obter bilhetes digitais ilegalmente é através de cartões de crédito comprometidos, ou seja, cartões que foram roubados ou clonados e que são usados para comprar bilhetes sem a autorização nem o conhecimento dos seus proprietários. Outras formas incluem comprometer contas de pontos de fidelidade, que são contas que acumulam pontos em troca de compras, *phishing*, que é uma técnica que consiste em enviar emails falsos para obter dados pessoais ou financeiros das vítimas, e comprometer contas de empresas, para realizar atividades maliciosas no nome destas, como comprar bilhetes com os fundos monetários destas ou para obter vantagens.

Depois de obter os bilhetes digitais ilegalmente, os burlões podem vendê-los na *darkweb*, onde estes são vendidos muito mais caros do que o seu preço justo, lucrando assim grandes quantidades de dinheiro [17].

Outra forma de fraude utilizando os bilhetes digitais, é que os burlões podem assumir o papel de vendedores de bilhetes *online* e parecerem que são legítimos. Eles podem criar *websites* falsos ou anúncios fraudulentos na internet,

oferecendo bilhetes digitais a preços baixos, aliciando os compradores. Uma pessoa que quer comprar um bilhete digital para a sua viagem, quer seja de autocarro, de avião, de comboio, procura sempre o preço mais barato. Se esta não tiver qualquer conhecimento sobre as fraudes, acaba por cair neste esquema, enviando dinheiro, dados de identificação, dados financeiros, em que no final poderá receber um bilhete digital inutilizável ou simplesmente não recebe nada em troca [17].

A bilhética digital está fortemente dependente da qualidade e da segurança das comunicações entre o cliente e a empresa. Se houver falhas nas comunicações ou nas aplicações, podem ocorrer problemas sérios que afetam a confiabilidade e a privacidade dos dados. Por exemplo, na compra e venda de bilhetes digitais, há muitas transações que envolvem dados sensíveis, como informações pessoais e financeiras, que podem ser interceptadas e divulgadas por agentes maliciosos, se a segurança na comunicação não estiver bem assegurada.

Para prevenir esses riscos, são usados protocolos criptográficos que garantem a integridade e a autenticidade dos dados, do servidor e do cliente. Por outro lado, o desenvolvimento desses protocolos é uma tarefa muito complexa e desafiadora, que requer um alto nível de conhecimento e rigor, pois qualquer erro ou lacuna nos protocolos pode ser explorado por fraudadores.

Um exemplo que ilustra a gravidade deste tema é o caso de Eric Donys Simeu, um homem que foi condenado a 4 anos e 10 meses de prisão nos Estados Unidos da América por utilizar técnicas de *phishing* para obter credenciais de empresas que pertencem ao GDS (*Global Distribution System* ou Sistema de Distribuição Global). Estas empresas do GDS fornecem serviços de reserva de viagens a agentes de viagem e sites relacionados a viagens, onde estes possuem credenciais exclusivas para autenticarem-se e emitirem bilhetes de viagens.

Eric Donys Simeu realizou esquemas de *phishing* direcionados aos clientes da Travelport e da Sabre, ambas pertencentes ao GDS, com o propósito de roubar as suas credenciais exclusivas. Com estas credenciais, ele acedeu aos servidores da Travelport e da Sabre para emitir vários bilhetes aéreos fraudulentos, avaliados num valor superior a 2 milhões de dólares. Estes bilhetes foram posteriormente vendidos ou utilizados por ele próprio e pelos seus cúmplices. Além da pena de prisão, Eric Donys Simeu também está obrigado a pagar uma indemnização de cerca de 162 mil dólares à Travelport [17], [18].

Em 2016, Karsten Nohl revelou que os sistemas de reserva de viagens eram inseguros e podiam ser acedidos com ataques de força bruta. Estes ataques consistiam em testar várias combinações de apelidos comuns e códigos alfanuméricos chamados PNR (*Passenger Name Records* ou Registo de Identificação do Passageiro). Os PNR são códigos únicos que identificam uma reserva de um passageiro e contêm informações como os detalhes do voo, o lugar reservado, informações de contacto, etc. [17], [19].

Após garantir o acesso aos sistemas de reserva, os atacantes podiam alterar datas e/ou destinos de voos, pedir reembolsos, ou viajar em nome das vítimas. Além disso, podiam usar as informações de contacto dos passageiros para enviar emails falsos a solicitar credenciais ou dados financeiros. Este tipo de ataque era possível porque os sistemas

de reserva não tinham um limite de tentativas de acesso e não registavam o histórico de acessos, tornando difícil a deteção de acessos não autorizados [17].

Outra forma de fraude é quando alguns empregados da empresa ou autoridades estão envolvidos em atividades fraudulentas ou corruptas. Por exemplo, eles podem fornecer informações privilegiadas a outras pessoas sobre as fiscalizações dos documentos de viagem, para evitar serem detetados, ou receber subornos para permitir a entrada de alguém sem documentos, com documentos falsos ou que tenha cometido algum tipo de fraude [17].

2.9.4. Consequências das fraudes de bilhetes digitais

A fraude de bilhetes digitais é um problema sério que afeta muitas empresas e pessoas. Como esses bilhetes são entregues quase de imediato após a compra, o risco de fraude online é difícil de reconhecer e prevenir. As consequências da fraude de bilhetes digitais podem ser:

- Danos à reputação da empresa. Os clientes que são vítimas de fraude podem culpar a empresa por vender bilhetes falsos, pois os fraudadores podem usar o seu nome e logotipo e criar *websites* idênticos para enganar. Isto acaba por prejudicar a imagem e a credibilidade da empresa no mercado.
- Perdas financeiras para a empresa e para os clientes. Os fraudadores lucram com o trabalho da empresa, sem pagar nada por ele, reduzindo a sua receita. Os clientes perdem dinheiro quando pagam por um bilhete que não recebem ou que não é válido;
- Reembolsos injustos para os clientes. A empresa pode ter de devolver o dinheiro aos clientes que foram enganados por fraudadores, mesmo que não tenha responsabilidade sobre a fraude. Isso gera um prejuízo adicional para a empresa.

No entanto, existem formas que ajudam a prevenir estas e outras consequências graves, que serão apresentadas a seguir [20].

2.9.5. Estratégias para prevenir as fraudes

2.9.5.1. Combater a corrupção

Como foi mencionado anteriormente, a fraude de bilhetes digitais pode envolver funcionários da empresa ou autoridades que se aproveitam do seu cargo para cometer atos fraudulentos ou corruptos. Estes infratores são difíceis de detetar e punir, por isso é essencial adotar medidas preventivas. Algumas ações que podem contribuir para reduzir este tipo de problema são: criar agências anticorrupção, oferecer salários justos aos colaboradores, criar mecanismos que desincentivem a corrupção, alterar o *staff* constantemente nos diferentes locais de trabalho, entre outras medidas [17].

2.9.5.2. Deteção e Cancelamentos de reservas fraudulentas

Para evitar fraudes, as reservas de viagens têm de ser submetidas a uma inspeção por sistemas de deteção de fraudes que atribuem uma pontuação de risco com base em vários fatores, por exemplo: os dados pessoais do cliente, a forma

como foi feita a reserva, o modo de pagamento, o destino e a antecedência da reserva. As reservas que obtêm uma pontuação elevada são aquelas que são consideradas suspeitas, logo, elas têm de ser inspecionadas e verificadas pelo departamento de segurança e com a entidade bancária do cartão de crédito para confirmar a legitimidade da reserva. Se houver confirmação de fraude ou incertezas, a reserva é cancelada ou é solicitado outro método de pagamento.

Além disso, as empresas de viagens devem reforçar a segurança das contas dos clientes, exigindo formas de identificação mais robustas, como a autenticação de dois fatores. Desta forma, evita-se que os bilhetes sejam reservados de forma fraudulenta por pessoas não autorizadas. As empresas de viagens também devem eliminar métodos de autenticação fracos, como PIN's e palavras-chave simples e curtas [17].

2.9.5.3. Partilha de informações

Uma forma de combater a fraude é partilhar informações entre as empresas de viagens e as entidades financeiras, que possuem dados sobre os seus clientes, as viagens, os pagamentos, etc. No entanto, nem todas as empresas partilham essas informações, muitas vezes por falta de confiança ou por receio de violar a lei de proteção de dados pessoais (RGPD). Isso dificulta a deteção e a prevenção de fraudes, como por exemplo, quando alguém usa um cartão de crédito roubado para comprar uma viagem, mas a empresa só sabe disso quando o dono do cartão reporta o acontecimento.

A partilha de informações pode ser uma solução eficaz para identificar e evitar fraudes, mas deve ser feita com cuidado e respeito pela privacidade dos clientes. As empresas devem obter um consentimento dos clientes para partilhar os seus dados pessoais ou ter uma razão válida para o fazer, como por exemplo, em caso de suspeita ou evidência de fraude. Nesses casos, as empresas podem ou devem partilhar as informações com outras entidades ou com as autoridades.

Adicionalmente, a partilha de informações pode ser benéfica para as empresas que são concorrentes entre si, pois podem ter uma visão mais ampla e atualizada das fraudes que ocorrem no mercado. Isso pode ajudá-las a detetar e a prevenir fraudes mais facilmente e eficazmente, o que acaba por beneficiar todos. Para isso, é importante que as empresas confiem nos dados que são partilhados e na entidade que gere a rede de partilha de informações.

Outro método que pode ajudar na deteção de fraudes é aumentar a colaboração com as autoridades, fornecendo-lhes mais dados [17].

2.9.5.4. Evitar a obtenção de bilhetes de forma fraudulenta

Como vimos no início, existem vários métodos de fraude para obter bilhetes de forma ilegal, tais como:

- Usar cartões de crédito comprometidos;
- Aceder aos sistemas de reserva sem autorização;
- Usar contas de pontos de fidelidade comprometidas;
- Usar contas de empresas comprometidas.

Quando uma conta é comprometida, os atacantes costumam mudar o email e/ou o número de telemóvel da conta. Por isso, é importante usar métodos que dificultem o acesso à conta e alterações de dados sem autorização. A autenticação por multifator pode ser uma solução, por exemplo, se o email for alterado, será enviado um código OTP (*One-Time Password* ou Palavra-passe de uso único) para o número de telemóvel e vice-versa. O acesso à conta pela aplicação no *smartphone* ou pelo *browser* também deve exigir uma autenticação por multifator. Por outro lado, a autenticação por multifator tem algumas limitações, por exemplo, a autenticação móvel requer uma conexão móvel, que pode implicar *roaming*, que é caro, ou *tokens* físicos, que são dispositivos pequenos, como chaves USB, cartões inteligentes ou dispositivos *bluetooth*, em que os titulares das contas têm de levar consigo e não podem os perder.

Atualmente, para invadir contas sem permissão, alguns atacantes usam programas automáticos que testam várias combinações de nomes de utilizador e palavras-passe até acertarem. Se as credenciais forem fracas, o programa pode descobri-las muito rápido. Para evitar isso, é preciso ter credenciais fortes ou usar CAPTCHAs (*Completely Automated Public Turing test to tell Computers and Humans Apart* ou Teste de Turing público completamente automatizado para diferenciar computadores e humanos), que são testes que aparecem em algumas páginas e pedem para identificar objetos em imagens, como bicicletas, escadas ou passadeiras. Estes testes servem para bloquear os programas dos atacantes, pois eles não conseguem reconhecer as imagens. Desta forma, os atacantes necessitam de mais esforço, mais tempo e mais dinheiro para tentar burlar esses sistemas de proteção.

Outra forma de combater a obtenção de bilhetes digitais ilegais é não avisar os passageiros fraudulentos com muita antecedência que os seus bilhetes foram cancelados, mas sim dar essa informação no momento do *check-in*. Assim, os fraudadores não teriam tempo de obter outro bilhete ilegal e seriam impedidos de viajar. Esta medida poderia reduzir as fraudes e o comércio ilegal de bilhetes digitais. Para as vítimas de fraudes, esta medida não faria muita diferença, pois teriam de comprar um bilhete novo de qualquer forma, já que o seu bilhete original foi roubado.

É importante reter que um bom mecanismo de prevenção e deteção de fraudes é um investimento, não um custo, pois permite evitar perdas financeiras e reputação [17].

2.9.5.5. Evitar a venda de bilhetes fraudulentos

Muitas vezes as autoridades concentram-se nas pessoas que possuem bilhetes digitais fraudulentos, no entanto não se pode esquecer a origem desse problema que é a venda e a revenda desses bilhetes. Provavelmente uma boa maneira de reduzir as fraudes nos bilhetes é concentrar-se mais nas pessoas que vendem e revendem esses bilhetes e não só apenas nas pessoas que viajam de forma fraudulenta. De forma simples, é melhor detetar e punir os reis do que os peões.

Outra forma de atrapalhar a venda de bilhetes fraudulentos no mercado negro online é criar um ambiente de desconfiança entre os compradores e os vendedores, chamado o "*Lemonising the market*". O objetivo "*Lemonising the market*" é fazer com que os compradores tenham medo de comprar bilhetes nos mercados negros online, pois eles podem ser falsos ou ilegais. Desta forma, eles podem preferir comprar nos sites oficiais que são garantidos serem

legítimos e seguros. Com isto, pode ajudar a reduzir o número e o lucro dos vendedores fraudulentos e dificultar os seus negócios.

As empresas normalmente têm listas negras que são listas que contêm registos de pessoas que estão proibidas de viajar com essa empresa por causa de atividades ilegais. O problema é que algumas dessas listas são limpas após algum tempo, o que permite às pessoas que estavam lá poderem viajar novamente com essa empresa, e os fraudulentos sabem disso. Uma solução para isto seria tornar as listas negras permanentes, ou seja, quem for apanhado a cometer fraudes, é inserido na lista e nunca mais poderá viajar por essa empresa, mesmo que ele tenha um bilhete completamente legal.

2.9.5.6. Combater o mercado negro ilegal

O mercado negro online ilegal é um dos meios usados para obter bilhetes de forma fraudulenta. Neste mercado, há vendedores, empresas falsas e comunidades que enganam os compradores com bilhetes ilegais. Uma forma de combater este problema é denunciar os *websites* que vendem estes bilhetes às entidades que os hospedam e pedir que removam os seus nomes de domínio. No entanto, nem todos os *websites* ilegais são fáceis de identificar. Alguns não imitam empresas reais e legítimas, o que torna mais difícil a sua deteção e remoção. Outros tentam copiar empresas reais e legítimas, esses são mais fáceis de serem denunciados, o que leva à desativação dos seus *websites* e ao bloqueio do seu acesso.

Algumas empresas têm o cuidado de criar *websites* como uma aparência profissional para parecer legítimo, adicionalmente elas podem chegar ao ponto de comprar licenças no Google Ads para promover os seus *websites* fraudulentos através de publicidade. Uma maneira de impedir que essas empresas comprem licenças no Google Ads é verificar se estão certificadas pelas entidades competentes que fazem a certificação das empresas na área dos transportes públicos [17].

2.9.6.Resumo dos métodos de fraudes

Como foi redigido anteriormente, com mais detalhe, existem diversas formas de fraude com os bilhetes digitais. Neste ponto, será sintetizado o que foi escrito e apresentado de forma simples os principais métodos de fraude neste tema. Desta forma, poderá contribuir para uma melhor compreensão para este problema. Alguns métodos de fraude são os seguintes:

- Usar cartões de crédito comprometidos para comprar bilhetes digitais;
- Comprometer contas de pontos de fidelidade para obter bilhetes digitais gratuitamente ou com descontos;
- Usar técnicas de *Phishing* para roubar dados sensíveis que podem ser usados para comprar bilhetes digitais ou aceder às contas das vítimas;
- Comprometer contas de empresas para comprar bilhetes com os fundos destas ou para obter vantagens;
- Criar *websites* ou anúncios fraudulentos na internet, que parecem legítimos, onde oferecem bilhetes digitais a preços reduzidos, aliciando os compradores;

- Fazer uma cópia do bilhete digital e transferi-lo para outro passageiro;
- Aceder aos sistemas de reserva com ataques de força bruta;
- Colaboradores ou autoridades usam informações privilegiadas ou subornos para facilitarem a fraude [17], [18], [19];

2.9.7. Resumo das estratégias de prevenção

À semelhança do ponto anterior, este ponto resume e apresenta de forma simples as estratégias para prevenir as fraudes de bilhetes digitais. Entre as estratégias, destacam-se:

- Criar agências anticorrupção;
- Oferecer salários justos aos colaboradores;
- Criar mecanismos que desincentivam a corrupção;
- Alterar o *staff* nos diferentes locais de trabalho constantemente;
- Submeter as reservas de viagens a uma inspeção com sistemas de deteção de fraudes;
- Reforçar a segurança das contas dos clientes e eliminar credenciais fracas;
- Partilhar informações entre as empresas e as entidades financeiras;
- Aumentar a colaboração com as autoridades;
- Usar métodos que dificultem o acesso à conta e alterações de dados sem autorização, como a autenticação por multifator;
- Usar credenciais fortes e/ou CAPTCHAs para dificultar os programas automáticos dos atacantes;
- Não avisar os passageiros fraudulentos com muita antecedência que os seus bilhetes foram cancelados;
- Concentrar as buscas nas pessoas que vendem e revendem bilhetes digitais ilegais;
- Criar um ambiente de desconfiança entre os compradores e os vendedores, chamado o "*Lemonising the market*";
- Tornar as listas negras permanentes;
- Denunciar os *websites* que vendem bilhetes fraudulentos às entidades que os hospedam e pedir que removam os seus nomes de domínio;
- Verificar se as empresas que compram licenças no Google Ads estão certificadas pelas entidades competentes na área dos transportes públicos [17].

2.9.8. Reflexão

Em conclusão, o desenvolvimento das novas tecnologias e a evolução das existentes possibilitaram o surgimento de novas formas de fraude digital, como a fraude de bilhetes digitais. Neste ponto 2.9., foram apresentadas algumas formas de fraude que podem ser praticadas com esses bilhetes digitais e a facilidade com que se pode obter acesso ilegal a eles. Este é um problema grave que ocorre com uma frequência crescente e tende a tornar-se mais complexo, eficaz e danoso. Por isso, é importante que as empresas adotem medidas preventivas e desenvolvam bons sistemas de deteção para combater as fraudes de bilhetes digitais. Além disso, é essencial que os consumidores desses bilhetes tenham a consciência e conhecimento suficientes para identificar casos suspeitos de fraude e agir de forma correta. No entanto, isso é um grande desafio, pois nem todas as pessoas têm a mesma capacidade de discernimento. Por fim, o importante é que todos estejam atentos e informados sobre as melhores práticas para evitar serem vítimas de fraude e que mantenham uma relação próxima com as autoridades competentes, denunciando os casos de fraude, para que sejam fiscalizados e punidos, garantindo assim uma maior segurança e confiança no mercado digital [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

2.10. Métodos de deteção de fraudes nos bilhetes digitais

Antes de apresentar esta secção, foi feita uma introdução ao conceito de fraude, aos tipos de fraudes em bilhetes físicos e digitais, às causas e consequências das fraudes, e às medidas preventivas que podem ser tomadas para reduzir a ocorrência de fraudes. Este ponto é dedicado à análise dos diferentes métodos de deteção de fraudes de bilhetes digitais, com o objetivo que mostrar como é possível identificar e combater as fraudes de bilhetes digitais.

2.10.1. Métricas e objetivos dos métodos de deteção de fraudes

Como já foi mencionado em diversas ocasiões, a fraude é um problema crescente e complexo que afeta várias áreas tecnológicas que fazem parte do nosso dia a dia, tais como as telecomunicações, as comunicações móveis, o *home banking* online, as compras online, entre outras. A evolução da tecnologia e da comunicação global incentiva a ocorrência de atividades fraudulentas, que geram grandes prejuízos para as empresas e para a sociedade. Desta forma, a deteção de fraudes tornou-se numa área de grande relevância e que requer investigação para combater as fraudes.

A deteção de fraudes tem como objetivo identificar as fraudes o mais cedo possível após a sua ocorrência. Para isso, são desenvolvidos e aprimorados mecanismos de deteção para defender as diferentes estratégias utilizadas pelos criminosos. No entanto, este desenvolvimento é dificultado pela falta de partilha de dados e da divulgação dos resultados, o que torna relevante a estratégia proposta no ponto 2.9.5.3., que consiste na partilha de informações para reduzir as fraudes. A deteção de fraudes baseia-se essencialmente na análise de grandes volumes de dados, tais como os dados registados e os comportamentos dos utilizadores, pois as fraudes são descobertas a partir de anomalias nos dados e padrões. A deteção de fraudes pode ser implementada por meio de diversas técnicas, como as heurísticas, etc. [21].

Um sistema de deteção de fraudes deve ser capaz de identificar corretamente as ações fraudulentas e legítimas, minimizando os erros de classificação.

Para avaliar o desempenho de um sistema de deteção de fraudes, são utilizadas três métricas principais: taxa de deteção, taxa de falso alarme e tempo médio de deteção. A taxa de verdadeiro positivo, também chamada taxa de precisão de deteção, é a percentagem de casos de fraude que são corretamente classificados como fraudulentos. A taxa de falso positivo, também chamada de taxa de falso alarme, é a percentagem de casos legítimos que são incorretamente classificados como fraudulentos. O tempo médio de deteção é o intervalo de tempo, em média, entre o momento em que uma ação fraudulenta ocorre e o momento em que é detetada pelo sistema de deteção.

O objetivo de um sistema de deteção de fraudes é maximizar a taxa de precisão de deteção, minimizar a taxa de falso positivo e reduzir o tempo médio de deteção. Isto acontece porque existe uma relação inversa entre a taxa de deteção e a taxa de falso positivo, ou seja, se uma aumenta, a outra diminui.

Além disso, é importante definir com rigor as métricas que são utilizadas para medir o desempenho do sistema, tendo em conta os objetivos [21].

2.10.2. Importância dos métodos de deteção de fraudes

Serão apresentados a seguir alguns fatores que demonstram a importância dos métodos de deteção de fraudes:

- **Redução da exposição de fraudes**: Os métodos de deteção de fraudes fortalecem os sistemas contra possíveis vulnerabilidades que possam ser aproveitadas por atividades fraudulentas, reduzindo assim as chances dos fraudadores realizarem atos ilegais.
- **Deteção confiável de fraudes**: Os sistemas de deteção de fraudes conseguem identificar atividades fraudulentas antes que elas causem prejuízos, aumentando assim o nível de controlo e de segurança, se eles forem bem desenvolvidos.
- **Aumento da confiança do cliente**: Os sistemas de deteção de fraude oferecem mais segurança para as empresas e para os seus clientes. Quando os clientes de uma empresa enfrentam poucos ou nenhum problema causado por fraudes, eles desenvolvem um grau de confiança na empresa e isso melhora a sua reputação e o seu crescimento.
- **Análise de dados não estruturados**: Por vezes, as fraudes são cometidas através da exploração destes dados não estruturados. Os sistemas de deteção de fraudes podem utilizar estes dados para identificar ou prevenir atividades fraudulentas [22]. Os dados não estruturados são informações que não seguem um modelo ou uma organização predefinida. Eles podem ser textos, datas, números, entre outros [23].
- **Deteta padrões ocultos**: Os sistemas de deteção de fraudes bem desenvolvidos conseguem identificar tendências, cenários e padrões ocultos, ou seja, que não são facilmente percetíveis.
- **Melhorias no desempenho da empresa**: Os sistemas de deteção de fraudes reduzem as atividades fraudulentas e, consequentemente, as perdas de receita. Assim, as empresas podem aumentar os seus lucros financeiros e também melhorar a eficiência e a qualidade das transações financeiras [22].

2.10.3. Desafios na implementação de sistemas de deteção de fraudes

Como foi referido anteriormente, um dos desafios no desenvolvimento de sistema de deteção de fraudes é a escassez de dados e de resultados partilhados.

Além disso, é difícil de reproduzir cenários reais de ataque ou avaliar o desempenho dos sistemas existentes, o que dificulta a sua implementação e melhoria.

Outra limitação é a falta de generalização dos sistemas de deteção de fraudes, que normalmente são construídos para um cenário específico e não podem ser facilmente adaptados a outros.

Outro problema dos sistemas de deteção de fraudes é que eles dependem dos perfis que foram previamente definidos. Se surgir uma nova forma de fraude, o sistema pode não ser capaz de identificá-la. Por isso, é necessário atualizar os sistemas com frequência e realizar manutenções, o que implica custos elevados e uma atenção especial por parte dos responsáveis.

Finalmente, outra limitação é a definição precisa das métricas e parâmetros do sistema, como os limites e os critérios de deteção, que devem ser ajustados dependendo dos objetivos [21].

2.10.4. Baseado em regras

A deteção de fraudes baseada em regras identifica atividades fraudulentas com base num conjunto de atributos incomuns, como registos de data e hora fora do padrão, números de conta atípicos, tipos de transação incomuns e valores discrepantes, entre outros critérios.

Este método de deteção de fraudes opera com base num conjunto predefinido de “regras” ou condições. Quando uma ou mais dessas condições são satisfeitas, há uma possibilidade de uma ou mais transações serem potencialmente fraudulentas. Dependendo de como o sistema é implementado, a transação pode ser bloqueada [24].

Na criação de um sistema de deteção, é comum implementar regras considerando diversos aspetos, dos quais alguns exemplos incluem:

- Localização - Este critério aplica-se em que as transações ocorrem fora da área de localização habitual do utilizador. Por exemplo, se um utilizador normalmente utiliza o seu passe de transporte regularmente na zona do Funchal, mas, de repente, há um uso regular na zona de Santana, isso pode ser um motivo de suspeita. Portanto, é importante que as regras abranjam esse fator.
- Frequência - Este critério aplica-se aos padrões de uso. Um utilizador que costuma utilizar o seu passe de transporte duas vezes por semana, mas, de repente, passa a usá-lo diariamente, ou seja, há um aumento significativo e incomum no uso do cartão, isso é um indicativo de suspeitas de fraude que deve ser considerado. Portanto, também a frequência é um fator a ter em conta na definição das regras.
- Remetente / Destinatário - Se um utilizador começa a receber pagamentos de contas recentemente criadas, especialmente em grandes quantidades, ou se alguém está criando várias contas com cartão de crédito a partir

do mesmo endereço IP num curto período de tempo e enviando dinheiro, pode levantar suspeitas de atividades fraudulentas. Portanto, a análise de remetentes e destinatários deve ser abordada nas regras de deteção [24].

2.10.4.1. Limitações

Portanto, podemos dizer que se uma transação não atende a nenhum dos critérios estabelecidos para identificar uma transação fraudulenta, ela é considerada como legítima. O sistema de deteção baseado em regras pode ser visto como um guarda, permitindo a passagem de algumas transações, enquanto sinaliza ou nega outras com base nas regras definidas, como mencionado anteriormente. No entanto, estes sistemas estão propícios a falhas, o que pode levar a ocorrência de falsos positivos e falsos negativos [24].

Além disso, os sistemas baseados em regras tornaram-se dispendiosas e ineficientes atualmente, uma vez que os fraudadores passaram a adotar estratégias mais complexas para burlar estes sistemas. É necessário ainda possuir um profundo entendimento do ambiente no qual o sistema irá operar, o que, por sua vez, pode acarretar desafios na extração e no resumo de regras eficazes. Em alguns cenários, não basta contar com um conjunto pequeno de regras, por vezes, é necessário lidar com uma grande quantidade delas, o que complica a gestão do conjunto de regras, pois requer mais capacidade computacional e torna a avaliação e a compreensão mais difícil [25].

A análise de fraudes baseada em regras pode ser desafiadora de gerir, uma vez que a configuração adequada das regras exige uma programação precisa, sendo um processo trabalhoso e demorado para abranger todas as possibilidades imagináveis de fraude. O surgimento constante de novos tipos de fraudes requer uma adaptação contínua das regras, a fim de incluir as possibilidades de fraude existentes, emergentes e futuras. Além disso, estes sistemas enfrentam desafios de escalabilidade e o seu desempenho diminui à medida que a quantidade de dados que precisam de ser analisados e o número de regras aumentam [21].

Embora os sistemas de deteção baseados em regras possam ser úteis e sofisticados, uma das suas principais limitações é a sua inflexibilidade, o que pode resultar num aumento significativo de casos falsos positivos [26].

Estes sistemas enfrentam um problema conhecido como “pontos cegos”, que se referem a áreas em que as regras existentes não conseguem fornecer cobertura, o que permite aos fraudadores explorarem as lacunas existentes nas regras. Os sistemas tradicionais dependem de regras definidas por serem humanos e exigem atualizações manuais em resposta às novas ameaças. Durante o período da atualização, pode haver tempo suficiente para que os fraudadores se aproveitam dessas lacunas. Daí a importância, ao desenvolver um sistema deste tipo, de considerar todos os cenários possíveis e realizar sessões de testes rigorosas. No entanto, como já foi mencionado, isso é uma tarefa difícil [24].

Estes sistemas são eficazes quando os agentes definem regras que sejam úteis e eficazes. Se as regras forem mal definidas ou incorretas, podem levar a falhas no sistema de deteção de fraudes e resultar num aumento de falsos positivos e falsos negativos, o que pode levar a transtornos. Portanto, é crucial ter plena consciência dos tipos de fraudes que podem ocorrer no ambiente em que o sistema opera, para isso é necessário ter um profundo conhecimento do projeto, e implementar as regras com precisão e rigor, uma vez que isso desempenha um papel decisivo na eficácia do sistema [24].

2.10.4.2. Vantagens

Os sistemas baseados em regras são eficazes na deteção de fraudes com base em critérios, reduzindo a necessidade de intervenção humana e, consequentemente, diminui os custos de segurança para as empresas [24].

Os sistemas baseados em regras são consideravelmente mais simples, e essa simplicidade contribui para uma boa eficiência operacional até um certo ponto, o que é um fator importante ao analisar transações individuais em tempo real. Estes sistemas analisam todas as transações e identificam potenciais fraudes com base em regras predefinidas, permitindo uma análise rápida das transações se o número destas e de regras for pequeno [24], [27].

Uma vez que estes sistemas operam com base em regras que são satisfeitas, eles são muito fáceis de interpretar. Isso significa que, se surgirem problemas, como falsos positivos ou falsos negativos, os desenvolvedores responsáveis podem identificá-los e solucioná-los de forma rápida. A facilidade de interpretação por parte deste modelo permite aos programadores criarem novas regras facilmente e rapidamente sempre que novos tipos de fraudes forem surgindo [24], [27].

Uma vez que estes sistemas funcionam com base em regras que são satisfeitas, eles são fáceis de interpretar. Isto significa que, se ocorrerem problemas, como falsos positivos ou negativos, os técnicos responsáveis podem identificá-los e corrigi-los rapidamente. Esta fácil interpretabilidade deste modelo permite aos desenvolvedores criem novas regras rapidamente de forma fácil sempre que um novo tipo de fraude é descoberta [24].

Em certos cenários, os sistemas de deteção de fraudes baseados em regras podem ser desenvolvidos e validados com facilidade, operando de forma rápida. A sua simplicidade permite uma rápida e fácil implementação, mas é importante referir que não são soluções que podem ser criadas e esquecidas, como mencionado anteriormente, elas devem ser continuamente atualizadas. A implementação destes sistemas é simples até um certo ponto, quando está perante projetos grandes e complexos que envolvem inúmeros tipos de fraudes, a sua implementação e interpretação pode revelar-se um verdadeiro desafio [24].

2.10.5. Modelos de Pontuação

Os modelos de pontuação são classificadores amplamente utilizados em transações devido à sua eficácia e facilidade operacional. Estes modelos utilizam técnicas que atribuem uma pontuação a cada transação, e geralmente quanto maior a pontuação, maior a probabilidade de a transação ser fraudulenta. Adicionalmente, estes tipos de modelos podem ser combinados com a classificação baseada em regras, onde envolve a atribuição de pesos ou pontuações a cada regra, que são então somados à medida que a transação satisfaz ou não às regras.

Neste modelo, é comum definir dois limites que determinam as ações a serem tomadas para cada transação:

- Se a pontuação for menor ou igual ao primeiro limite, a transação é aprovada, pois a probabilidade de ser fraudulenta é baixa.
- Se a pontuação estiver entre os dois limites, a transação é sinalizada para análise manual detalhada pela equipa técnica.

- Se a pontuação for maior que ambos os limites, a transação é bloqueada, pois é altamente provável que seja fraudulenta.

Este modelo torna o processo de deteção de fraude mais escalável, uma vez que, embora todas as transações sejam pontuadas, apenas algumas delas requerem uma análise minuciosa e individual pelos técnicos. Isto ajuda a otimizar os custos de recursos humanos, permitindo uma alocação mais eficiente dos técnicos para cada transação. No entanto, este tipo de modelo apresenta alguns desafios, o que inclui a necessidade de manter os modelos atualizados ao longo do tempo e garantir uma alta taxa de precisão nas previsões, pois uma revisão e manutenção contínua requer custos elevados [26].

2.11. Reflexão Global

Neste capítulo, o foco principal foi explorar as definições e métodos de deteção proporcionadas pela literatura existente, através de artigos e outras fontes. No ponto 2.1, que aborda a fraude em geral, foram apresentados o conceito de fraude, a distinção entre fraude ocupacional e organizacional, os elementos que a compõem, as razões pelas quais a sua deteção e visualização são tarefas difíceis, alguns tipos e formas de fraude, além de um exemplo verídico que evidencia a sua gravidade. Finalmente, foram apresentados alguns métodos para a sua prevenção.

No ponto 2.8, o objetivo foi ser mais específico, concentrando-se nas fraudes de bilhetes. Aqui, foram abordadas a origem e os tipos de fraude de bilhetes no contexto dos transportes públicos, seguido de definições mais específicas para cada tipo de fraude. Além disso, explorou-se os motivos pelos quais as pessoas cometem fraude na área dos transportes públicos, destacando a importância de combater essas práticas e apresentando os principais prejuízos e possíveis métodos de prevenção. Por fim, realizou-se uma análise comparativa de alguns sistemas de controlo de bilhetes.

O ponto 2.9 tornou-se ainda mais específico, abordando as fraudes de bilhetes digitais. Iniciou-se com uma breve introdução aos bilhetes digitais no contexto dos transportes públicos, seguido da apresentação dos motivos que levam as pessoas a cometer fraudes nessa área. Foram abordadas algumas formas e exemplos de fraudes de bilhetes digitais, seguidas das consequências dessas práticas. Por último, foram apresentadas estratégias que podem ajudar na ocorrência de fraudes em bilhetes digitais.

No ponto 2.10, o objetivo foi apresentar os métodos utilizados na deteção de fraudes de bilhetes digitais. Este ponto começou com a apresentação das métricas e objetivos dos métodos em geral, destacando a sua importância e os desafios associados ao desenvolvimento de sistema de deteção de fraudes de bilhetes digitais, Em seguida, foram apresentados os métodos utilizados nestas áreas, como os sistemas de deteção baseados em regras, com as suas vantagens e desvantagens. Foi abordado ainda os sistemas de deteção que incorporam sistemas de pontuações para determinar a presença de fraude e que relações têm com os sistemas baseados em regras.

Adicionalmente, é importante salientar que idealmente seria apresentar casos de estudo específicos para cada método abordado. Contudo, lamentavelmente, durante o período designado para a pesquisa e redação deste capítulo, não foi possível encontrar artigos que apresentassem casos de estudos de deteção de fraude de bilhetes digitais e

eletrónicos. Durante um período dedicado à pesquisa, foram encontrados vários casos de estudo relacionados aos métodos em questão, no entanto, estes estavam sempre direcionados para outros temas como: fraudes financeiras, fraudes em telecomunicações, deteção e prevenção fraude no comércio eletrónico, deteção e prevenção em pagamentos eletrónicos, proteção da privacidade dos clientes, fraudes com cartões de crédito, intrusões informáticas fraudes com cartões de crédito, entre outros.

Os artigos identificados sobre as fraudes na área dos transportes públicos, na sua maioria, centravam-se na proteção da privacidade dos clientes ou em medidas preventivas, em vez de abordarem a deteção de fraude de bilhetes digitais. Além disso, verificou-se a existência de artigos que abordavam de maneira abrangente o tema, apresentando métodos de deteção, mas não incluíam qualquer caso de estudo prático. Foram utilizadas as seguintes palavras-chave na pesquisa desses artigos: "*Electronic Ticketing*, *Fraud detection techniques*, *Public transport*, *Ticketing security*, *Transport*, *Fraud*, *Fraud detection*, *Rule-based approach*, *Customer profiles*, *Bus*, *Train*, *Ticket*, *Ticket fraud*".

Adicionalmente, foi tentado obter informações através de e-mail enviados para entidades como a SATA, Carris, Serviços Municipalizados de Transportes Urbanos de Coimbra, Sociedade de Transportes Coletivos do Porto, CP e TAP. Era esperado que pudessem contribuir para este estudo, fornecendo informações sobre os métodos que utilizam na deteção de fraudes de bilhetes digitais e uma avaliação do desempenho das soluções adotadas, em que incluiria a eficácia, a taxa de falsos positivos e falsos negativos, e se as soluções atendiam satisfatoriamente aos seus objetivos. Infelizmente, a SATA recusou a colaborar, a CP solicitou um comprovativo de inscrição no mestrado, mas não deu mais seguimento, enquanto as restantes entidades não responderam ao pedido.

Este capítulo desempenhou o papel de uma introdução ao tema das fraudes de bilhetes digitais, estabelecendo uma base teórica e conceptual para os capítulos subsequentes. No próximo capítulo, será abordado especificamente o problema que pretendemos resolver com este trabalho, a metodologia proposta para o seu desenvolvimento e a avaliação.

3. Definição e Análise do Problema

Este capítulo descreve em detalhe o problema específico enfrentado pela empresa Horários do Funchal, no que diz respeito à deteção de fraudes em bilhética eletrónica. Apresenta a proposta de abordagem baseada em regras, as ferramentas consideradas para a implementação do sistema, e o método de avaliação da solução. O capítulo estabelece a ponte entre a teoria e a implementação prática.

3.1. Sistema de bilhética da Horários do Funchal

A Horários do Funchal pretende desenvolver um sistema de bilhética integrada num futuro próximo. Como salientado anteriormente, especialmente no capítulo 1, os bilhetes são suscetíveis a fraudes, o que pode resultar em perdas financeiras consideráveis. Para evitar esses riscos e minimizar possíveis impactos negativos, é essencial adotar medidas proativas no combate desse problema. Desta forma, o projeto de suporte a esta dissertação consiste na elaboração e implementação de um sistema para a deteção de fraudes, cujo benificiário é a empresa Horários do Funchal. O principal objetivo é assegurar que o sistema seja capaz de identificar e classificar com precisão e eficiência os casos de fraude. Para compreender a natureza do projeto e definir os principais tipos de fraudes que o sistema deverá identificar e classificar, foram realizadas reuniões com o responsável pelo sistema de bilhética.

Embora a Horários do Funchal já tenha solucionados alguns casos, é crucial manter a consciência que é possível que esses casos possam ocorrer novamente.

Portanto, é sempre importante ter em mente os seguintes casos:

- A deteção de comportamento anómalo, como o uso de um passe/título por múltiplas pessoas, foi abordada pela HF (Horários do Funchal) por meio da implementação de uma solução chamada de "*passback*". Essencialmente o *passback* impede que uma pessoa empreste o seu passe/título para outra entrar no mesmo autocarro imediatamente após o primeiro uso. Tipicamente o *passback* possui um período de validade determinado, bloqueando o acesso do passe/título por um intervalo específico após o seu uso inicial.
- Outra situação crítica envolve o uso de um passe/título sem que o saldo correspondente seja debitado, constituindo uma potencial fraude. Neste cenário, trata-se de uma falha no sistema, portanto, é crucial que o desenvolvedor assegure uma integridade absoluta do sistema, uma vez que qualquer falha pode resultar em perdas financeiras. Esta forma de fraude não apenas implica o utilizador final, mas também o próprio desenvolvedor do software. Sendo assim, destaca-se a necessidade de abordar ambos os lados da moeda, exigindo uma verificação rigorosa para garantir que o sistema tenha sido desenvolvido em conformidade com todas as regras e especificações necessárias. Portanto, trata-se de uma fraude de dois sentidos: o desenvolvedor, devido a um erro que resultou em perda de fundos, e o cliente, que se beneficiou da falha para obter vantagens indevidas nas suas viagens.
- Embora a validação à entrada e à saída do autocarro seja uma solução considerada para estes problemas, ela apresenta problemas significativos. Um dos principais problemas é a determinação do preço a ser

praticado à entrada. Imagine-se um cenário em que o passe tem um saldo de 2€, e a viagem até ao destino custa 3,20€. A validação à entrada precisa ser aceite neste caso, no entanto, o passageiro pode optar por marcar a segunda validação apenas ao chegar ao destino e vai embora ou, eventualmente, esquecer-se de validar. Isso levanta a questão: qual é o preço que deve ser cobrado à entrada? Além disso, a implementação desses sistemas requer mais equipamentos, tornando-se mais propenso a avarias, como avarias no validador de entrada, no validador de saída ou em ambas as situações. A própria HF experimentou esse tipo de abordagem, mas, devido a desafios operacionais, optou por abandoná-la. Um problema adicional é que este sistema ainda é suscetível a situações em que o passageiro valida a entrada e, no meio da viagem, valida a saída, possibilitando a continuação da viagem sem custos adicionais. Estes são alguns casos reais que já ocorreram nos Horários do Funchal, destacando as limitações desta abordagem.

- Os códigos QR são frequentemente utilizados para validar bilhetes eletrónicos móveis, a possibilidade de cópia requer uma análise abrangente dos registos em busca de anomalias associadas a códigos que não correspondem a vendas genuínas. No entanto, a Horários do Funchal propõe a geração de códigos QR aleatórios representando bilhetes por períodos específicos, sendo renovados a cada nova compra. Esses códigos serão de uso único, inviabilizando a validação por terceiros, caso sejam copiados.

Considerando o que foi dito anteriormente, o objetivo do projeto pode ser delineado da seguinte maneira:

- O principal objetivo deste projeto é assegurar que cada validação de bilhete seja classificada como legítima ou fraudulenta. A deteção de tais inconsistências pode ser realizada por meio de um sistema antifraude ou de alertas. Por outras palavras, é necessário criar um sistema de deteção baseado em regras que analise um conjunto de dados, em que caso algum satisfaz a uma ou várias regras, é considerado suspeito ou, em caso extremo, uma fraude. Este sistema opera com regras fixas e é particularmente adequado para lidar com casos heurísticos, como descritos neste objetivo. As heurísticas são mais utilizadas em situações em que a simplicidade e a rapidez na tomada de decisões são fundamentais. No contexto da validação de bilhetes, a aplicação de heurísticas irá simplificar o processo, adotando uma abordagem direta entre "corresponde ou não corresponde" a uma venda, eliminando considerações intermediárias.

Ao longo desta definição do problema, foram mencionadas algumas regras, tanto heurísticas quanto não heurísticas, de uma forma não exaustiva. Para proporcionar maior clareza, apresentam-se de forma mais estruturada as regras que se podem aplicar.

Algumas regras podem ser:

- O mesmo passe ou bilhete não pode ser utilizado duas vezes consecutivas ou simultaneamente.
- Não pode haver reentradas de bilhetes de bordo, pois estes são de utilização única.
- A validação de um bilhete é permitida apenas durante um intervalo de tempo específico para o titular.

- A segunda validação de um bilhete ou passe na mesma viagem e no mesmo autocarro não é permitida.
- Assegurar que os bilhetes e os passes têm validades adequadas para o seu tipo.
- Verificar que em cada entrada há um consumo da viagem no passe ou bilhete.
- Examinar validações de criança em intervalos de tempo incomuns, por exemplo durante a madrugada.
- Analisar se os bilhetes possuem viagens restantes no momento da validação.
- Assegurar que os bilhetes multiviagens por dias têm os minutos restantes de viagem como também deve ter em conta possíveis excessos de minutos permitidos.
- Bilhetes e passes urbanos não podem ser utilizados em operadores interurbanos quanto na rede interurbana.
- Assegurar que a tarifa correta foi aplicada nas validações dos passes e bilhetes.
- Verificar se os bilhetes ou passes validados estão atribuídos ao grupo correto.

3.2. Proposta de abordagem

Nesta secção, será apresentado o método que será utilizado para abordar o problema definido e analisado anteriormente. A escolha deste método é crucial, pois o sucesso deste projeto depende inteiramente das suas capacidades de combater eficazmente o problema, gerar resultados de alta qualidade e confiáveis, e fundamentar conclusões concretas. Por outras palavras, é imperativo selecionar o método mais adequado para o problema em questão, a fim de produzir resultados satisfatórios em todos os aspetos.

Dada a natureza do problema previamente definido, analisado e para o que se pretende, considera-se que a abordagem mais eficaz para a deteção de fraudes sejam os sistemas baseados em regras. Os sistemas de deteção baseados em regras são eficazes na identificação de casos simples, diretos e claros de fraude, que são evidentes e facilmente percetíveis.

Por outro lado, é importante notar que o desempenho destes sistemas pode ser comprometido em situações de fraude mais sofisticadas, onde as manipulações podem ser mais elaboradas e difíceis de serem detetadas por meio de regras predefinidas. No entanto, por agora pretende-se apenas detetar os casos mais simples e diretos, desta forma, a utilização de um sistema de deteção baseado em regras atende adequadamente às normas heurísticas estabelecidas no ponto 3.1.

Apesar de, conforme discutido na secção 2.10.4, os sistemas baseados em regras serem considerados, em muitos contextos, dispendiosos e ineficientes, a sua adoção neste trabalho é justificada por diversas razões práticas e contextuais.

Em primeiro lugar, o sistema baseado em regras permite uma total transparência e auditabilidade, facilitando a compreensão e aceitação por parte da Horários do Funchal. Em segundo lugar, a integração com ferramentas já conhecidas (como Pentaho e MySQL) evita a necessidade de infraestruturas ou competências técnicas adicionais.

Além disso, os padrões de fraude observados são relativamente repetitivos e estruturados, o que favorece a eficácia de uma abordagem determinística, sobretudo numa fase exploratória. Por fim, este sistema funciona como um protótipo funcional, que poderá futuramente evoluir para soluções mais sofisticadas baseadas em *machine learning*.

Assim, a escolha desta abordagem não se deve a uma preferência teórica, mas sim a uma decisão estratégica e pragmática, alinhada com os objetivos e recursos disponíveis.

3.2.1.Ferramentas para a deteção de fraudes baseada em regras

3.2.1.1. SEON

A SEON é uma empresa especializada na prevenção e deteção de fraudes online, disponibilizando uma variedade de ferramentas que atendem às necessidades contemporâneas, para realizar essa missão. Esta empresa oferece ferramentas modulares de fácil integração e de utilização, são aplicáveis a diversas indústrias e podem ser personalizadas para cada setor empresarial.

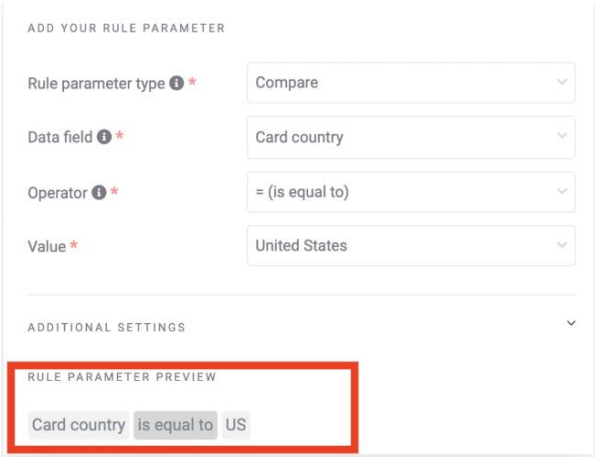


Imagem 4 - Criação da regra [28]

No âmbito da deteção de fraudes baseada em regras, a SEON disponibiliza uma ferramenta que permite a criação de regras personalizadas através de uma interface e de fácil utilização. As imagens 4 e 5 ilustram esse procedimento, tendo em conta que a imagem 5 demonstra a capacidade de criar regras adicionais relacionadas com outras [28].

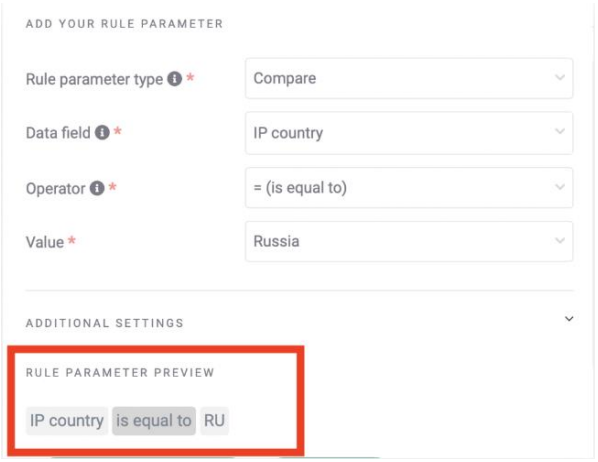
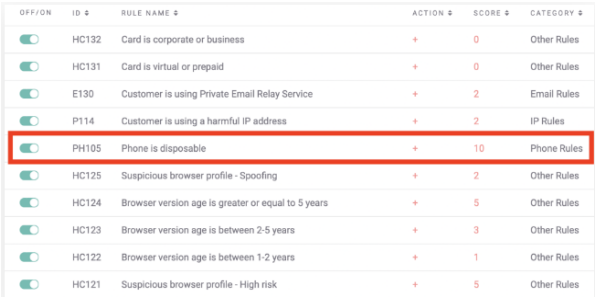


Imagem 5 - Criação da regra [28]

Esta ferramenta ainda permite facilitar a gestão das regras personalizadas criadas de forma intuitiva e simples, como se pode evidenciar na imagem 6 [28].

Imagem 6 - Gestão de regras [28]

Complementarmente, é possível configurar sistemas de pontuação para as regras estabelecidas, como demonstrado na imagem 7 [28].



OFF/ON	ID	RULE NAME	ACTION	SCORE	CATEGORY
	HC132	Card is corporate or business	+	0	Other Rules
	HC131	Card is virtual or prepaid	+	0	Other Rules
	E130	Customer is using Private Email Relay Service	+	2	Email Rules
	P114	Customer is using a harmful IP address	+	2	IP Rules
	PH105	Phone is disposable	+	10	Phone Rules
	HC125	Suspicious browser profile - Spoofing	+	2	Other Rules
	HC124	Browser version age is greater or equal to 5 years	+	5	Other Rules
	HC123	Browser version age is between 2-5 years	+	3	Other Rules
	HC122	Browser version age is between 1-2 years	+	1	Other Rules
	HC121	Suspicious browser profile - High risk	+	5	Other Rules

Imagem 7 - Sistema de pontuação para as regras [28]

Conforme mostrado na imagem, a coluna “pontuação” diz respeito à pontuação atribuída a cada regra, em que no final esses pontos são calculados para determinar o nível de risco da ação.

Consequentemente, poderá ser necessário estabelecer limites com base nas pontuações de fraude, exemplificado na imagem 8 [28].

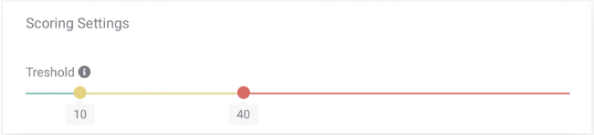


Imagem 8 - Configuração dos limites de pontuação [28]

Como se pode verificar, a codificação de cores é utilizada para interpretar as pontuações, sendo que o verde (0-10 pontos) indica a aprovação das ações, o amarelo (10-40 pontos) indica uma revisão necessária e o vermelho (40+ pontos) indica a negação das ações. É importante destacar que estes limites não são estáticos, a SEON enfatiza a flexibilidade destes parâmetros, permitindo ajustes conforme as necessidades específicas [28].

Adicionalmente, a SEON disponibiliza uma ferramenta que permite a configuração de *Velocity Rules* para a deteção de fraudes, ou seja, estas buscam compreender o comportamento do utilizador ao observar as ações realizadas ao longo de um determinado período. Por outras palavras, por meio dessas regras, é possível, por exemplo, monitorizar o número de tentativas de login num intervalo de 5 minutos. Um exemplo prático é sua aplicação no procedimento de login, onde elas ajudam a evitar que os fraudadores utilizem métodos, como ataques de força bruta ou ferramentas automatizadas para descobrir as credenciais, para comprometer a autenticação. A configuração desta funcionalidade é exemplificada na imagem 9 [28].

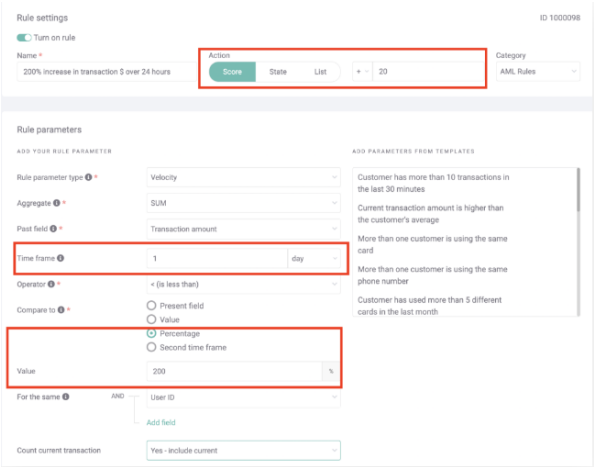


Imagem 9 - Configuração de velocity rules [28]

Neste caso, a ferramenta lida com o aumento significativo nos gastos (mais de 200%) num período de 24 horas. Quando tal situação ocorre, a regra predefinida é acionada, podendo ainda ser atribuída uma pontuação. As *velocity rules* tornam-se úteis para identificar comportamentos suspeitos, como o uso consecutivo de bilhetes digitais num curto espaço de tempo [28].

A SEON oferece uma ferramenta especializada na deteção de fraudes, permitindo a aplicação de regras personalizadas para determinar a autenticidade de uma ação. Por outro lado, apesar de todos os benefícios, é uma solução paga e com preços consideráveis. Embora ofereça um plano gratuito, dependendo do projeto, pode ser insuficiente, como a restrição a apenas 10 regras personalizadas, uma taxa de apenas 2 queries por segundo, entre outras [28].

3.2.1.2. Pentaho

O Pentaho é um software de código aberto para a integração e análise de dados. Com esta ferramenta é possível desempenhar tarefas de ETL (*Extraction*, *Transformation* e *Load*), *reporting*, OLAP e mineração de dados (*data mining*). Adicionalmente, o Pentaho foi considerado um dos melhores do seu segmento.

Este sistema é capaz de realizar análises de *big data* utilizando *data wearhouses*, base de dados, etc [29].

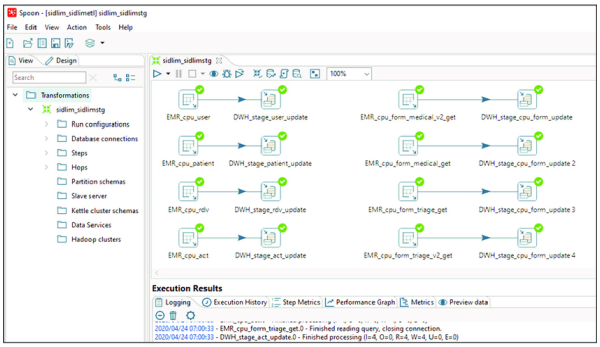


Imagem 10 - Interface do Pentaho Data Integration (Kettle)

[30]

O Pentaho Data Integration é uma das ferramentas que compõem o Pentaho, e que pode ser usada para a deteção de fraudes baseada em regras. Esta ferramenta permite conectar-se a uma *Data Wearhouse* e usar um *step* de *Modified* JavaScript *Value* que possibilita inserir código JavaScript na transformação. Com este código, pode-se aplicar lógica condicional, cálculos ou outras operações nos dados de entrada, e criar novos campos de saída com os resultados. Por exemplo, para detetar transações fraudulentas, pode-se usar critérios como o valor, a origem, o destino, a distância, o tempo, a frequência, entre outros, e usar o *step* de JavaScript para verificar esses critérios e atribuir uma pontuação de risco ou classificar os dados como fraudulentos ou não. Posteriormente é possível visualizar os resultados num relatório ou num *dashboard*, usando as ferramentas adequadas do Pentaho para esse fim.

Também o Pentaho é capaz de ler e escrever dados diretamente de uma base de dados, ou seja, o Pentaho é capaz de ir lendo as linhas de dados da base de dados e ao mesmo tempo fazer todos os processos necessários e depois escreve na mesma base de dados, ou noutra, ou num ficheiro, entre outros. Além disso, o Pentaho também permite executar *scripts* de *queries* SQL diretamente na base de dados o que pode ser muito útil para verificar os critérios e atribuir uma classificação dos dados.

No entanto, é necessário ter em conta que o Pentaho requer um intervalo de tempo para executar as transformações e análises, ou seja, à medida que a complexidade das transformações e dos dados aumentam, o intervalo de execução também pode aumentar. Desta forma, é essencial otimizar as transformações para melhorar a eficiência do processamento.

Embora ainda não tenham sido encontradas implementações documentadas especificamente voltadas para deteção de fraudes com base em regras no Pentaho, os recursos apresentados indicam que esta aplicação é viável. A exploração e validação dessas possibilidades será parte integrante do desenvolvimento do projeto.

3.2.1.3. Talend

Assim como o Pentaho, o Talend é uma ferramenta de código aberto destinada à integração de dados, em que ambas as plataformas possibilitam a realização de tarefas ETL (Extração, Transformação e Carregamento), permitindo extrair dados de diversas fontes, transformá-los e carregá-los em destinos como *data wearhouses*, bases de dados, ficheiros, entre outros.

Tal como o Pentaho, o Talend também possui a capacidade de criar *pipelines* de dados e executá-los em diferentes ambientes, incluindo o Apache Spark e as mais recentes tecnologias em nuvem. Desta forma, o Talend consegue integrar uma ampla variedade de tipos de dados provenientes de diversas fontes, tanto localmente quanto em ambientes de nuvem. Adicionalmente, esta plataforma oferece a capacidade de criar regras em Java para a deteção de fraudes, assim como o Pentaho, isto significa que ambas as plataformas permitem desenvolver componentes personalizados em Java e integrá-los às transformações de integração de dados [30].

Embora haja uma falta de informações específicas possa gerar alguma incerteza, é importante mencionar que há um caso em que o Talend foi aplicado para a deteção de fraudes baseado em regras. Desta forma, foi conduzida uma entrevista com um profissional que já utilizou o Talend para esse fim, e que confirmou que a ferramenta possibilita a criação de regras de deteção de fraudes em Java, apresentando algumas semelhanças com a abordagem do Pentaho.

Portanto, pode-se concluir que o Talend e o Pentaho são similares em termos de funcionalidades, incluindo o processo de criação de regras para a deteção de fraudes.

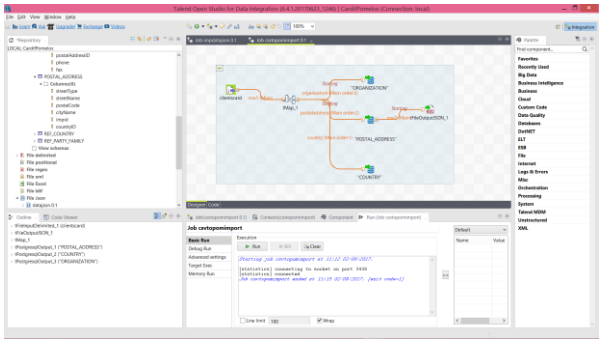


Imagem 11 - Interface do Talend [31]

3.2.1.4. Fatores de seleção da ferramenta

A SEON oferece um plano gratuito com limitações significativas, e os seus planos pagos são consideravelmente dispendiosos, o que a exclui das opções viáveis. O Pentaho e o Talend são duas ferramentas que apresentam funcionalidades e características semelhantes, onde pode haver algumas pequenas diferenças entre elas. No entanto, a Horários do Funchal manifesta uma clara preferência pelo Talend, devido à sua experiência prévia com esta ferramenta. Além disso, a organização já possui uma base de conhecimento e de experiência na implementação de regras para a deteção de fraudes usando o Talend. No entanto, a viabilidade do Talend e, consequentemente, a sua utilização para desenvolver o sistema de deteção de fraudes baseado em regras, foi prejudicada. No dia 31 de janeiro de 2024, esta ferramenta deixou de ser de código aberto, tornando-se uma solução paga, devido à diminuição da adoção pela comunidade e à falta de contribuições para o projeto. Desta forma, torna-se imperativo utilizar o Pentaho, que é uma alternativa bastante semelhante ao Talend.

3.2.2. Reflexão

O sistema baseado em regras proporciona uma abordagem equilibrada e suficiente, que é capaz de lidar tanto com normas heurísticas previstas, que abrangem casos mais simples e diretos, mas que também é um sistema fácil de manter, fácil de perceber e fácil de corrigir se houver algum erro.

As ferramentas apresentadas como a SEON, o Pentaho e o Talend, destacam-se pelas suas capacidades distintas na criação e na gestão de regras para a deteção de fraudes. A SEON, por exemplo, oferece uma interface intuitiva para a configuração de regras, incluindo a atribuição de pontuações para avaliar o nível de risco. No entanto, é importante reconhecer que apesar das suas vantagens, esta ferramenta é uma solução que pode representar um problema de acessibilidade, especialmente em projetos com recursos financeiros limitados.

O Pentaho e o Talend, sendo ferramentas de código aberto, proporcionam uma maior flexibilidade e personalização na criação de regras de deteção de fraudes. Ambas as ferramentas demonstram a capacidade de integrar dados de várias fontes e aplicar lógica condicional para identificar casos suspeitos. A comparação entre estas duas plataformas revela semelhanças nas suas funcionalidades, destacando a viabilidade de escolher entre elas com base nas preferências e nos requisitos específicos do projeto.

Desta forma, a abordagem proposta não apenas apresenta soluções para o problema em questão, mas também descreve algumas das melhores ferramentas disponíveis, não só para resolver o problema, mas também otimizar a sua resolução, proporcionando resultados sólidos e concretos.

3.3. Método de avaliação da solução

Para garantir a eficácia da solução desenvolvida, será necessário segmentar o conjunto de dados em diferentes partes, com variações quantitativas, de modo a simular um ambiente o mais realista possível de operação. O objetivo central é verificar se o sistema de deteção baseado em regras consegue identificar e classificar de forma precisa as validações fraudulentas. Além disso, será considerada a inserção intencional de dados anómalos no conjunto a ser testado, a fim de simular cenários adicionais de fraude.

Para uma validação rigorosa da eficácia das regras implementadas, recomenda-se desativar temporariamente todas as outras regras, exceto aquela que está sendo testada no momento. Este procedimento garante que qualquer classificação atribuída pelo sistema possa ser diretamente associada à regra em teste, permitindo uma avaliação precisa do seu desempenho. A partir dessa análise, será possível determinar se as validações fraudulentas foram corretamente identificadas e classificadas.

No caso de classificações incorretas, como uma validação legítima sendo classificada como fraude ou uma validação fraudulenta sendo marcada como legítima, será necessário rever a lógica da regra. Estas falhas indicam possíveis erros na formulação das regras, que deverão ser ajustadas para atribuir uma classificação correta.

Um dos aspetos cruciais na avaliação do desempenho do sistema é a medição do tempo necessário para classificar todas as validações presentes no conjunto de teste. Esta análise é essencial para entender o impacto que o tamanho da amostra tem sobre o desempenho do sistema. É esperado, e considerado normal, que amostras maiores resultem num tempo de processamento mais elevado, uma vez que o sistema precisará de realizar mais operações de classificação. Este fator deve ser cuidadosamente monitorizado, pois qualquer aumento excessivo no tempo de classificação pode comprometer a eficiência do sistema num cenário de produção.

Além disso, será realizada uma análise da densidade de fraudes, definida como a relação entre o número de fraudes detetadas e o tamanho total da amostra. Esta métrica será importante para identificar padrões comportamentais de fraude ao longo do tempo, permitindo uma análise mais aprofundada. Com base na densidade, poderemos, por exemplo, verificar se há variações significativas na ocorrência de fraudes durante dias específicos, como fins de semana em comparação com dias úteis, e analisar tendências mensais ou sazonais. Este tipo de estudo pode revelar picos de atividade fraudulenta em determinados períodos, auxiliando na criação de estratégias para lidar com potenciais ameaças.

Somente após o sistema estarem dentro dos requisitos definidos é que poderá ser considerado prontos para a implementação em prodição. Até lá, é essencial continuar otimizando e corrigindo as regras existentes, tanto no aspeto da lógica quanto da eficiência de execução, a fim de garantir resultados precisos e cumprir com os limites de desempenho aceitáveis.

O desenvolvimento de uma solução robusta, confiável e altamente eficiente na identificação de fraudes é o principal objetivo deste projeto.

3.4. Viabilidade da proposta

O presente projeto de dissertação, desenvolvido em parceria com a Horários do Funchal, contará com o fornecimento crucial de uma extensa amostragem de dados históricos. A organização comprometeu-se a disponibilizar cerca de 17 milhões de registos, os quais foram devidamente anonimizados, garantindo a confidencialidade das informações sensíveis. Estes dados são operacionais e já se encontram no formato tabular, simplificando significativamente o processo de desenvolvimento.

A relevância deste fornecimento de dados transcende a mera disponibilidade, pois representa um alicerce fundamental para o desenvolvimento e sucesso do projeto de tese. A amplitude da amostragem não apenas enriquece a análise, mas também potencializa a eficácia da deteção de fraudes. Por outras palavras, a vasta quantidade de dados proporciona a oportunidade de explorar diversos cenários e testar o sistema de deteção de fraudes em múltiplos casos. Esta abordagem diversificada contribuirá para a robustez do sistema, permitindo a identificação de fraudes e de comportamentos anómalos de forma eficaz.

Assim, a contribuição da Horários do Funchal não apenas facilita a condução do projeto, mas também eleva significativamente o potencial de um sistema de fraudes de alta qualidade e adaptável a diversas situações de fraudes no mundo real.

No que diz respeito à ferramenta selecionada, o Pentaho foi escolhido como a principal solução de *Business Intelligence* (BI) e análise de dados para este projeto. O Pentaho, sendo uma plataforma de código aberto amplamente adotada, destaca-se não só pela sua robustez técnica, mas também pela comunidade ativa que continuamente contribui para o seu crescimento e melhoria contínua. A Pentaho Community Home desempenha um papel crucial, servindo como um ponto de encontro que oferece um ecossistema colaborativo, através do qual os utilizadores podem partilhar experiências, discutir soluções técnicas, troca de ideias, *feedback*, documentação, etc. Adicionalmente, a vasta e acessível documentação do Pentado facilita a aprendizagem e implementação das ferramentas, permitindo aos desenvolvedores compreender de maneira simples o funcionamento interno do *software*. Isto acelera o desenvolvimento de solução um pouco mais complexas, como sistemas de deteção de fraudes, simplificando a criação de *pipelines* de dados e automação de processos, o que, em última análise, resulta numa implementação mais eficiente e com menos erros.

Outro ponto a destacar é a flexibilidade do Pentaho, que permite a integração com outras tecnologias e bases de dados, tornando-o uma ferramenta adequada para projetos com uma grande análise de dados como o que está a ser desenvolvido nesta tese.

É importante ainda realçar que as ferramentas fornecidas pelo Pentaho podem simplificar o desenvolvimento dos sistemas de deteção, uma vez que algumas delas automatizam procedimentos.

Concluindo, a viabilidade desta ferramenta não se baseia apenas na sua robustez técnica, mas também na comunidade ativa que a sustenta, refletindo um compromisso contínuo com a expansão e melhorias constantes.

4. Desenvolvimento da Solução Proposta

Este capítulo documenta o processo de desenvolvimento da solução de deteção de fraudes. Apresenta a arquitetura do sistema, as ferramentas utilizadas (Pentaho, MySQL), a estrutura da base de dados, e as transformações ETL desenvolvidas. São também descritas as 13 regras definidas para deteção de padrões anómalos. O objetivo é demonstrar a implementação prática da proposta e preparar os testes de validação.

4.1. Sistema

4.1.1. Pentaho

O sistema de deteção de fraudes baseado em regras foi desenvolvido utilizando a ferramenta Pentaho Spoon, também conhecida como Pentaho Data Integration. A escolha desta ferramenta foi motivada não só pelo foi descrito no ponto 3.4. mas também pela familiaridade prévia, pela sua intuitividade e simplicidade de uso, aliadas à capacidade de realizar uma variedade de operações nos campos e nos dados. O Pentaho Data Intregation, como já foi dito, funciona como um fluxo de dados, onde diversos *steps* podem ser aplicados ao longo do processo para executar diversas tarefas. Adicionalmente, cada um destes componentes é configurado de acordo com os requisitos específicos do processo, proporcionando flexibilidade e adaptabilidade ao fluxo. Esta abordagem é modular e configurável, facilitando a construção de processos de ETL (*Extract*, *Transformation* e *Load* – Extração, Transformação e Carregamento) complexos e personalizados de forma a atender às necessidades especificas de cada projeto.

4.1.2. MySQL

Durante o desenvolvimento do projeto do sistema de deteção de fraudes baseado em regras, uma ferramenta adicional, o XAMPP, foi incorporada para oferecer o uso de uma base de dados MySQL local, embora não estivesse inicialmente prevista. Esta inclusão revelou-se útil devido à necessidade de manipular e criar novos dados e campos a partir daqueles que já existem. O MySQL foi selecionado devido à sua eficácia em lidar com grandes volumes de dados tabulares, facilitando a criação de novos campos de forma mais eficiente. Além disso, há uma maior familiaridade com o mesmo em comparação ao uso de JavaScript para a manipulação de ficheiros CSV. Dessa forma, o processo em JavaScript exigiria um esforço e um tempo consideravelmente maiores, enquanto, com o MySQL, a experiência prévia permite que essa criação e manipulação dos dados seja realizada com maior agilidade e menos esforço. Da mesma forma, o MySQL acelera o processo de classificação dos dados, uma vez que as classificações são enviadas para a base de dados através de um *query* com argumentos.

Adicionalmente, o Pentaho possui uma integração robusta com o MySQL, exigindo apenas a instalação de um *driver* e a configuração da conexão com a base de dados. Em relação à execução das *queries*, estas podem ser facilmente escritas utilizando o *step* “Execute SQL script”, que envia a instrução para o servidor.

É crucial destacar que estes novos campos foram criados com o propósito específico de facilitar a criação e classificação por parte das regras do sistema de deteção de fraudes.

A base de dados está sendo utilizada pelo sistema de deteção de fraudes baseada em regras, com a possibilidade de ser também usada por outro sistema que venha a ser desenvolvido no futuro.

4.2. Base de dados MySQL

Esta secção destina-se a fornecer uma descrição de cada campo presente na tabela, a fim de garantir que em fases posteriores do presente trabalho, ao mencionar o nome de um campo ou colunas da tabela, o leitor tenha uma compreensão clara do seu significado e propósito. Dado o grande número de colunas na tabela, não é possível apresentar todas elas numa única imagem, como ilustrado na imagem 22.

DataHora	Carreira	Chapa	DHIniViagem	Activo	Descricao	DataHoraRegisto	Linha	Codigo	TipoEvento	Veiculo	PosValid	NSConc	NSValid	IDValid	VarPercurso	SentidoPerc	Viagem	ParagINIOrd	ParagFIMOrd	ParagemEntradaOrd	TempoViagem
2022-12-31 15:27:44	15	12	2022-12-31 15:24:58	A	Passe Social Invalidez I	2022-12-31 15:46:19	1035005530	356485184	3	254	1	87000426	34000848	1	0	2	6	1	28	5	
2022-12-31 12:59:59	N/A	N/A	0000-00-00 00:00:00	A	Passe Social - 12 meses - Interurbano - F7	2022-12-31 15:46:21	1035005531	356485200	3	1943	1	87000438	3221347402	0	0	-1	-1	-1	-1	-1	
2022-12-31 13:00:09	N/A	N/A	0000-00-00 00:00:00	A	Passe Social - 12 meses - Interurbano - F7	2022-12-31 15:46:21	1035005532	356485201	3	1943	1	87000438	3221347402	0	0	-1	-1	-1	-1	-1	
2022-12-31 13:03:15	N/A	N/A	0000-00-00 00:00:00	A	Passe Social II - Interurbano - F3	2022-12-31 15:46:21	1035005533	356485202	3	1943	1	87000438	3221347402	0	0	-1	-1	-1	-1	-1	
2022-12-31 13:58:07	N/A	N/A	0000-00-00 00:00:00	A	Passe Social - 12 meses - Interurbano - F7	2022-12-31 15:46:21	1035005534	356485219	3	1943	1	87000438	3221347402	0	0	-1	-1	-1	-1	-1	
2022-12-31 14:30:41	N/A	N/A	0000-00-00 00:00:00	A	Passe Social Invalidez II - Interurbano - F7	2022-12-31 15:46:22	1035005535	356485224	3	1943	1	87000438	3221347402	0	0	-1	-1	-1	-1	-1	
2022-12-31 15:28:00	4	19	2022-12-31 15:23:00	A	Livre Trânsito	2022-12-31 15:47:31	1035005536	356485259	3	409	1	87000467	34000649	1	0	2	14	1	27	11	
2022-12-31 14:47:06	4	19	2022-12-31 14:47:05	A	Passe Social Invalidez I	2022-12-31 15:47:34	1035005537	356485277	3	409	1	87000467	34000649	1	0	1	13	1	28	1	
2022-12-31 14:42:27	13	24	2022-12-31 14:41:55	A	Passe Social Invalidez I	2022-12-31 15:49:56	1035005538	356485354	3	234	1	87000690	34000617	1	0	1	6	1	29	1	

Imagem 12 - Tabela da base de dados

As colunas da tabela ilustrada na Imagem 12 são as seguintes:

- **DataHora** – Representa a data e hora da validação.
- **Carreira** – Representa a carreira onde a validação ocorreu.
- **Chapa** – A chapa representa um conjunto de viagens.
- **DHIniViagem** – Indica a data e hora de início da viagem.
- **Ativo –** Indica se o passe ou bilhete está ativo quando os dados são retirados da base de dados.
- **Descricao** – Descreve por escrito o tipo de objeto validado.
- **DataHoraRegisto** – Representa a data e hora em que o registo foi inserido na base de dados.
- **Linha e Codigo** – Representa um identificador único para diferenciar os diferentes registos.
- **TipoEvento** – Indica se a validação é uma entrada (1) ou reentrada (3).
- **Veículo** – Número do veículo onde ocorreu a validação.
- **PosValid** – Indica o resultado da validação, sendo (1) aceite e (0) recusada, podendo ser devido a problemas como má leitura ou bilhete/passe inválido.

- **VarPercurso** – Indica a presença de uma variante no percurso. O valor (0) corresponde ao percurso normal, enquanto diferentes de zero denotam outras variantes.
- **SentidoPerc** – Indica o sentido da viagem. O valor (1) representa o sentido de ida ou subida, enquanto o valor (2) corresponde ao sentido de volta ou descida.
- **Viagem** – Representa o número de viagens dessa chapa. Por exemplo, o valor (15) nesta coluna indica que o passageiro validou o seu passe ou bilhete na décima quinta viagem dessa chapa.
- **ParagINIOrd** – Denota o número da primeira paragem, que será sempre 1.
- **ParagFIMOrd** – Representa o número da última paragem, por exemplo, se for 28, isso indica que a viagem compreende um total de 28 paragens.
- **ParagemEntradaOrd** – Indica o número corresponde à paragem na qual o passageiro embarcou.
- **TempoVuagemRestanteMins** – Refere-se ao tempo restante que o passageiro tem para viajar. Este campo é mais relevante para os bilhetes pré-comprados de uso múltiplo.
- **ZonaCorrente** – Indica a zona onde a validação ocorreu, por exemplo, o valor (23) corresponde ao Funchal.
- **NSCartaoHI, NSCartaoLO** – Eles representam partes do número de identificação atribuído ao cliente, bem como ao passe ou bilhete pré-comprado, excluindo os bilhetes de bordo.
- **CounterValueBefore** – Indica o número de viagens restantes antes da validação, sendo relevante principalmente para os bilhetes pré-comprados.
- **CounterValueAfter** – Representa o número de viagens restantes após a validação, sendo significativo principalmente para os bilhetes pré-comprados.
- **Titulo** – Refere-se ao número que identifica um tipo de passe ou bilhete.
- **GrupoTitulo** – Este campo refere-se à agrupação dos títulos, sendo o Grupo 1 designado para os bilhetes e o Grupo 2 para os passes.
- **NumViagens –** Indica o número de viagens atribuído ao passageiro conforme especificado no seu passe ou bilhete.
- **NumDias –** Indica o número de dias atribuído ao passageiro conforme especificado no seu passe ou bilhete.
- **NumMeses –** Indica o número de meses atribuído ao passageiro conforme especificado no seu passe ou bilhete.
- **TipoTarifa –** Denota o tipo de tarifa aplicada, por exemplo, o tipo (1) corresponde a tarifas para criança, o tipo (2) refere-se a tarifas para séniores e o tipo (3) é aplicável a reformados, pensionistas e pessoas com invalidez.
- **Operador –** Identifica o operador onde se deu a validação, sendo que o valor (1) corresponde a Horários do Funchal e o valor (5) representa o operador interurbano.
- **Rede –** Identifica a rede na qual a validação ocorreu, onde o valor (1) e (3) corresponde à rede Urbana, o valor (256), (257) e (259) correspondem à rede interurbana.
- **NSCartao –** Este campo é uma concatenação dos campos NSCartaoHI e NSCartaoLO, com o objetivo de facilitar a verificação de múltiplas validações para o mesmo NSCartao. Além disso, serve como base para a

criação dos campos subsequentes. A decisão de concatenar os campos for tomada para simplificar a verificação, tornando mais eficiente o processo de comparação de apenas um campo em vez de dois.

- **Classificacao –** Representa a classificação atribuída pelas regras após a análise dos dados. As classificações possíveis incluem 'Legítimo', 'Fraude', 'Suspeito' ou 'Erro do sistema'.
- **PrimeiraValidacao –** Regista a data e hora da primeira validação de cada NSCartao.
- **UltimaValidacao –** Regista a data e hora da validação anterior a atual para cada NSCartao. A sua função principal é facilitar a deteção de validações simultâneas, comparando os registos de DataHora com o campo UltimaValidacao.
- **CooldownEnd –** Regista a data e hora em que termina o período durante o qual não é permitida uma segunda validação do mesmo NSCartao. Ele é derivado do campo DataHora, no qual a data e o tempo são somados, a fim de evitar validações muito próximas entre si.
- **UltimoVeiculo –** Indica o número do veículo onde ocorreu a validação anterior para cada NSCartao. Este campo, juntamente com o UltimoDHIniViagem e o UltimoPosValid, é utilizado para verificar se ocorreram validações no mesmo autocarro e na mesma viagem.
- **UltimoDHIniViagem –** Regista a data e hora do início da viagem da validação anterior para cada NSCartao.
- **UltimoPosValid –** Indica se a validação anterior para cada NSCartao foi aceite ou não.
- **UltimoCooldownEnd –** Regista o término do período durante o qual não pode ocorrer uma segunda validação do mesmo NSCartao após a validação anterior. Ele permite verificar se a data e hora da validação atual ocorrem antes da data e hora de término do período de espera da validação anterior para esse NSCartao. Por exemplo, consideremos que a primeira validação ocorreu às 22:24:12 do dia 14/03/2024, o que implica que o CooldownEnd é definido como 22:29:12 do dia 14/03/2024. No entanto, o campo UltimoCooldownEnd permanece vazio, pois não há validações anteriores para esse NSCartao. Na validação subsequente, que ocorreu às 00:24:55 do dia 16/03/2024, o CooldownEnd foi calculado como 00:29:55 do dia 16/03/2024. Desta vez, o campo UltimoCooldownEnd regista o tempo de 22:29:12 do dia 14/03/2024, indicando a data e hora do término do período de espera da validação anterior para esse NSCartao.
- **ValidacaoHora –** É um campo que guarda o valor de "DataHora" mas no formato HH:MM.

4.3. Transformações

A construção do sistema ocorre num ambiente denominado de transformações, onde uma variedade de *steps* são implementados e executados para obter os resultados desejados. Este ambiente de transformações não apenas simplifica o processo de construção, mas também viabiliza a adaptação às mudanças e à contínua evolução do sistema conforme as exigências e necessidades emergentes. Além disso, as transformações proporcionam uma interface praticamente gráfica, o que facilita significativamente o desenvolvimento, minimizando a necessidade de programação, que por sua vez pode ser mais vantajoso.

A divisão do sistema em três transformações distintas – limpeza de dados, manipulação de dados e classificação de dados – foi concebida para aderir às melhores práticas da engenharia de dados. Esta abordagem permite uma

organização clara e eficiente dos processos envolvidos na transformação dos dados, garantindo uma maior facilidade na modularidade e na escalabilidade do sistema. Adicionalmente, esta separação é crucial para assegurar uma execução sequencial e ordenada das transformações. Garante-se que a segunda transformação inicie somente após a conclusão da primeira, e que a terceira tenha início somente após o término da segunda. Esta sincronização é essencial uma vez que a transformação subsequente depende dos resultados da anterior. Ao estabelecer esta dependência clara entre as transformações, evitam-se problemas de integridade e de consistência nos dados. Esta abordagem também contribui para uma melhor gestão de erros e problemas, uma vez que cada transformação pode ser monitorizada e tratada individualmente. Assim, esta separação não apenas promove a qualidade dos resultados, mas também facilita a identificação e correção de eventuais problemas durante o processo de transformação dos dados.

É importante referir que as três transformações podem ser configuradas para serem executadas automaticamente, uma após a outras, assim que a anterior é concluída, apenas de estarem separadas. Esta configuração oferece uma camada adicional de automação e conveniência ao processo.

As transformações no Pentaho Data Integration representam o ambiente no qual se constrói o fluxo composto pelos seus *steps*, que são responsáveis por executar diversas tarefas sobre os campos de dados, como já referido. Uma vez que a transformação esteja finalizada, é possível executá-la para iniciar o movimento dos campos e dos dados ao longo do fluxo criado, permitindo que os *steps* desempenhem as suas funções.

As transformações oferecem uma ampla gama de *steps* para a limpeza, manipulação e transformação de dados, incluindo filtragem, junção, ordenação, agregação, eliminar campos desnecessários, conversão do tipo de dado (inteiro para *string*, por exemplo), executar scripts, etc. Após a aplicação destes *steps*, os valores dos campos podem ser sobescritos ou novos campos de fluxo podem ser criados preservando os valores originais.

É importante ressaltar que as transformações operam de forma sequencial, o que significa que os *steps* são executados numa ordem específica.

Após esta breve introdução sobre as transformações será agora discutido, detalhadamente, cada uma que compõem o sistema de deteção de fraudes baseado em regras, bem como os seus respetivos *steps*. Esta análise permitirá compreender melhor como as diferentes transformações e os seus componentes contribuem para o funcionamento e a eficácia do sistema, fornecendo uma visão abrangente e detalhada do processo de classificação dos dados por meio de regras.

4.3.1. Transformação *Data Cleaning*

A finalidade desta transformação é eliminar todos os campos desnecessários, duplicados ou redundantes, lidar com valores nulos, atribuir valores iniciais aos novos campos criados, remover os segundos e milissegundos do tempo, eliminar linhas idênticas e criar um campo novo resultante da concatenação entre o NSCartaoLO e NSCartaoHI, o qual será posteriormente convertido para formato inteiro. Em seguida, os campos e dados serão enviados para a base de dados, onde, uma vez concluído, a próxima transformação será executada automaticamente.

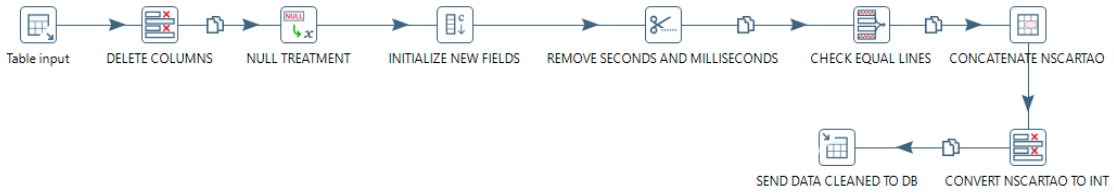


Imagem 13 - Fluxo da transformação *data cleaning*

- **Table Input** – Este step é responsável por extrair os dados da tabela da base de dados, juntamente com as suas colunas, e importá-los para o fluxo. Este *step* oferece a capacidade de selecionar a base de dados, a tabelas e escolher as colunas ou campos que serão incluídas no fluxo.
- **DELETE COLUMNS** – Este *step* tem a função de remover as colunas ou campos que não são pertinentes ao processo de classificação. Isso deve-se ao facto de existirem valores irrelevantes que não serão utilizados nas regras de classificação, ou existirem colunas ou campos com valores duplicados ou à presença de redundância.
- **NULL TREATMENT** – Este *step* tem a responsabilidade de substituir todos os valores nulos por um valor que denote ser inválido entre os valores restantes. Por exemplo, -1 para números inteiros, 0000-00-00 00:00:00 para datas, "N/A" para *strings* e 0 para booleanos. Esta prática é essencial, em primeiro lugar, como parte do processo de limpeza de dados, em segundo lugar, o tratamento de valores nulos é crucial para fases futuras, como o desenvolvimento de sistemas de deteção de fraudes com outras tecnologias e abordagens, e, em terceiro lugar, porque a omissão desse tratamento resultaria em erros ao enviar os dados para a base de dados nesta transformação.
- **INITIALIZE NEW FIELDS** – Este step tem a função de definir os novos campos criados e atribuir um valor inicial a eles. Esta etapa é crucial, pois o Pentaho Data Integration detetaria a presença de colunas adicionais na tabela da base de dados em comparação com os campos recebidos no step de envio dos campos e dados para a base de dados. Isso resultaria num erro, indicando que, por exemplo, a tabela possui 20 colunas, mas apenas 15 foram recebidas, deixando as outras 5 sem atribuição.
- **REMOVE SECONDS E MILLISECONDS** – Este step tem como objetivo remover a parte dos segundos e milissegundos das horas, uma vez que, para os propósitos das regras e análises realizadas, a precisão até ao segundo é suficiente. Os milissegundos representam um intervalo de tempo extremamente pequeno, que não oferece vantagens significativas em termos de precisão para as operações e análises conduzidas neste contexto. Num caso específico, extrair a hora e o minuto do campo "DataHora" e armazenar num novo campo denominado "ValidacaoHora" não requer precisão até ao segundo. Para mais detalhes, consulte a descrição da quinta regra no ponto 4.3.3., na secção referente ao *step* DATA CLASSIFICATION, e no ponto 8.3., na secção referente ao teste da quinta regra.
- **CHECK EQUAL LINES** – Este *step* visa eliminar linhas duplicadas, garantindo que cada linha no conjunto de dados seja única. Embora campos como "Linha" e "Codigo" sejam únicos e automaticamente incrementados em cada linha, é improvável que ocorram linhas duplicadas, no entanto, este *step* serve como

uma precaução adicional. Em caso de *bugs*, erros ou outras situações inesperadas que possam resultar em linhas duplicadas, este *step* é capaz de corrigir essas ocorrências.

- **CONCATENATE NSCARTAO** – Este *step*, conforme mencionado anteriormente, concatena os campos NSCartaoHI e NSCartaoLO para simplificar a verificação de múltiplas validações associadas ao mesmo NSCartao. Além de facilitar essa verificação, este novo campo serve como base para a criação de campos subsequentes. A decisão de concatenar os campos foi tomada visando simplificar e tornar mais eficiente o processo de comparação, substituindo a necessidade de analisar dois campos distintos por apenas um.
- **CONVERT NSCARTAO TO INT** – Este *step* realiza a conversão do campo criado no *step* anterior para um tipo de dado inteiro, visto que após a concatenação, o campo resultante é armazenado como uma *string*. Dado que os campos NSCartaoHI e NSCartaoLO são inteiros, é apropriado converter o campo derivado deles para o mesmo tipo de dado inteiro. Esta conversão garante a consistência e a conformidade dos dados, facilitando operações futuras que dependem do uso deste campo.
- **SEND DATA CLEANED TO DB** – Este *step* é responsável por enviar todos os campos e dados do fluxo para a base de dados. Esta transferência assegura que as informações estejam prontamente disponíveis numa plataforma mais eficiente para lidar com grandes volumes de dados, para serem utilizadas nas transformações subsequentes. É fundamental realizar a limpeza dos dados antes de enviá-los, garantindo assim que apenas dados limpos e relevantes sejam armazenados. Isso evita o desperdício de espaço com campos e informações desnecessárias, contribuindo para uma base de dados mais eficiente em termos de desempenho e utilização de recursos.

4.3.2. Transformação Data Manipulation

O objetivo primordial desta transformação é fazer uso dos dados fornecidos para realizar operações por meio de *queries* SQL, resultando na criação de novos dados para os campos criados. Estas operações abrangem desde a recuperação de informações de registos anteriores até a realização de cálculos para gerar os dados complementares. Esta metodologia possibilita a geração de dados adicionais que, por sua vez, podem ser empregues pelas regras estabelecidas para analisar casos que não seriam possíveis com base apenas nos dados originais. Esta abordagem amplia a capacidade do sistema, permitindo uma avaliação mais abrangente e precisa das informações disponíveis.

Imagem 14 - Fluxo da transformação data manipulation

- **SET COOLDOWN END** – Este *step* é responsável de executar *queries* que calculam o fim do período em que uma nova validação para o mesmo NSCartao não é permitida. Esta estratégia visa prevenir a ocorrência de validações muito próximas no tempo para o mesmo NSCartao. Naturalmente, existe algumas exceções,

como a presença de dois ou mais validações consecutivas devido a leituras incorretas do sistema, ou casos em que o passageiro acidentalmente passou o bilhete ou o passe duas vezes, ou até mesmo situações em que o sistema realizou duas leituras. No entanto, a verificação dessas exceções é da responsabilidade das regras de classificação dos dados.

- A primeira *query* essencialmente obtém o campo DataHora e adiciona 5 minutos, salvando o resultado no campo CooldownEnd. Este campo indica a data e hora em que expira o período em que não é permitida outra validação para o mesmo NSCartao.
- A segunda *query* é essencial, pois recupera o CooldownEnd da validação anterior e o armazena no campo UltimoCooldownEnd. Esta ação é útil, pois permite que as regras verifiquem se a validação atual para um NSCartao está dentro do intervalo entre o UltimoDataHora da validação anterior e o UltimoCooldownEnd dessa mesma validação.

- **GET/UPDATE NEW FIELDS** – Este *step* executa várias *queries* para recuperar dados de validações anteriores para o mesmo NSCartao.
  - A primeira *query* obtém a DataHora da primeira validação de cada NSCartao e armazena no campo PrimeiraValidacao. Embora este dado não seja atualmente utilizado pelas regras em vigor, ele facilita a identificação da data e hora da primeira validação de um determinado NSCartao. Dessa forma, ao analisar os dados, não é necessário realizar *queries* adicionais para encontrar o primeiro registo de um NSCartao específico. No entanto, é importante destacar que executar uma *query* para identificar a primeira validação de cada cliente no meio de um grande volume de registos resulta numa perda de desempenho durante a execução da transformação responsável pela manipulação dos dados.
  - A segunda *query* obtém a DataHora da validação anterior para cada NSCartao e armazena no campo UltimoDataHora. Este dado é útil para verificar se existem duas validações muito próximas para o mesmo NSCartao, em conjunto com o campo UltimoCooldownEnd. É importante observar que se o valor for "0000-00-00 00:00:00" indica que não há validações anteriores e, portanto, o registo atual representa a primeira validação para esse NSCartao.
  - A terceira *query* obtém o Veiculo da validação anterior para cada NSCartao e o armazena no campo UltimoVeiculo. Esta informação é valiosa para verificar se houve duas ou mais validações no mesmo autocarro, na mesma viagem e considerando que a validação anterior foi aceite. Isso é avaliado em conjunto com os campos UltimoDHIniViagem e UltimoPosValid, discutidos anteriormente. As *queries* relacionadas a esses campos serão apresentadas imediatamente após este texto. É importante notar que se o valor for "-1", indica que não houve validações anteriores e, portanto, não houve viagens de autocarro associadas a esse NSCartao. Desta forma, o registo atual representa a primeira validação para esse cliente e, portanto, o primeiro autocarro em que ele embarcou.
  - A quarta *query* obtém a DHIniViagem da validação anterior para cada NSCartao e o armazena no campo UltimoDHIniViagem. Este dado é essencial para determinar se houve ou mais validações no

mesmo autocarro, na mesma viagem e considerando que a validação foi aceite. Esta análise é feita em conjunto com os campos UltimoVeiculo e UltimoPosValid. É relevante observar que se o valor for "0000-00-00 00:00:00", indica que não houve validações anteriores e, portanto, o passageiro não embarcou em nenhuma viagem. Assim sendo, o registo atual representa a primeira validação e a primeira viagem desse NSCartao.

- A quinta *query* obtém o campo PosValis da validação anterior para cada NSCartao e o armazena no campo UltimoPosValid. Este dado é crucial para determinar se ocorreram duas ou mais validações no mesmo autocarro, na mesma viagem e considerando que a avalidação anterior foi aceite. Esta avaliação é feita em conjunto com os campos UltimoVeiculo e UltimoDHIniViagem, relações que já foram notadas anteriormente. É relevante reter que se o valor for "-1", indica que não houve validações anteriores e, portanto, não é possível afirmar se esta foi aceite ou não para esse NSCartao. Nesse caso, o registo atual representa a primeira validação para esse passageiro.

4.3.3. Transformação Data Classification

A principal finalidade desta transformação é extrair os dados contidos na tabela guardada na base de dados e aplicar um conjunto predefinido de regras, as quais irão atribuir uma classificação conforme a sua satisfação ou não. Posteriormente, as classificações resultantes são então enviadas de volta para a base de dados. Este processo automatizado não apenas simplifica a análise dos dados, mas também agiliza o processo de classificação.

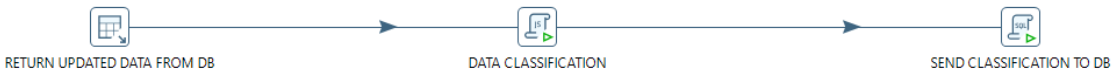


Imagem 15 - Fluxo da transformação data classification

- **RETURN UPDATED DATA FROM DB** – Cabe a este *step* a responsabilidade de reimportar os dados e as respetivas colunas da base de dados para o fluxo, já contendo os campos novos atualizados com os dados necessários para serem utilizados pelas regras. Adicionalmente, ele oferece a flexibilidade de selecionar as colunas ou campos a serem importados no fluxo.
- **DATA CLASSIFICATION** – Este *step* é responsável por armazenar e executar todas as regras destinadas a classificar os dados. No início de cada regra, é realizada uma verificação inicial para validar a legitimidade da classificação de uma transação. Caso a classificação seja legítima, a validação prossegue para a próxima regra. No entanto, se a validação já tiver sido identificada como fraude, erro do sistema, etc., não é necessário aplicar mais regras, o que resulta numa melhoria no desempenho do sistema. Além disso, esta verificação prévia impede que uma validação previamente classificada como fraude, por exemplo, seja incorretamente considerada legítima pelas regras subsequentes, pois, como será observado, se a condição de uma regra não for verificada, ela automaticamente classifica a validação como legítima. Esta abordagem não apenas otimiza o processo de classificação, mas também evita potenciais transtornos decorrente de classificações incorretas.

- A primeira regra verifica a presença de duplicidades de validações simultâneas para o mesmo NSCartao. Esta regra realiza uma comparação entre a data e hora da validação atual e a data e hora da validação anterior associada ao mesmo cliente. Esta comparação visa determinar se ambas as validações ocorrem simultaneamente. Por outras palavras, analisa a coluna "DataHora" em relação à coluna "UltimaValidacao" para verificar se possuem datas e horas idênticas. Caso seja confirmada esta coincidência, será identificado como um caso de fraude, por outro lado, estra regra classificará a validação como legítima. É de notar que os bilhetes de bordo estão excluídos desta regra, pois, como já foi dito e será visto na regra seguinte, são de utilização única. Portanto, não faz sentido incluí-los nesta regra. É relevante salientar que este tipo específico de fraude é improvável de ocorrer nos dados que serão analisados por esta regra. Isto deve-se à dificuldade de realizar duas validações simultâneas no exato mesmo segundo. Esse desafio decorre do método atual de validação de títulos, realizado por meio de um validador presente no interior dos autocarros, ou seja, os passageiros precisam manualmente efetuar a validação dos seus títulos ao entrar no veículo, portanto, é inerente à intervenção humana.
- A segunda regra identifica casos de reentrada de bilhetes de bordo, considerando que estes são de uso único, por outras palavras, um bilhete de bordo sendo ele de utilização única não pode ser reutilizado. Para isso, a regra analisa o número de identificação associado ao bilhete de bordo, ou o número correspondente ao bilhete de bordo para adultos, verificando se o tipo de evento registado é uma reentrada (TipoEvento = 3). De notar que este tipo específico de fraude dificilmente irá ocorrer nos dados que serão analisados por esta regra. Isto deve-se aos mecanismos que a Horários do Funchal possui, que antecedem este, e eu bloqueiam reutilizações do bilhete de bordo no momento da validação.
- A terceira regra analisa se o TempoViagemRestateMins e o CounterValueBefore estão ambos a zero. Esta regra visa examinar os bilhetes pré-comprados para viagens de adultos, bem como os bilhetes pré-comprados para grupos de adultos, juntamente com os bilhetes pré-comprados para viagens de crianças e os bilhetes pré-comprados para grupos de crianças. O objetivo é verificar se esses bilhetes ainda possuem minutos de viagem e viagens disponíveis. Em caso negativo, considera-se que ocorreu uma fraude. Por outras palavras, a regra utiliza o número de identificação vinculado a estes bilhetes pré-comprados para verificar a disponibilidade de minutos e de viagens. Este procedimento é justificado pela duração padrão de uma viagem nestes bilhetes, que é de 60 minutos. Contudo, muitos destes bilhetes podem conter múltiplas viagens (CounterValueBefore >= 1), por exemplo, um bilhete pré-comprado pode armazenar três viagens, sendo que cada uma delas deduzirá gradualmente o tempo disponível à medida que o passageiro utiliza o serviço de transporte público. Quando o tempo disponível para uma viagem é esgotado, o sistema da Horários do Funchal desconta automaticamente uma viagem no total restante. Portanto, chegará a um momento em que o bilhete pré-comprado não terá mais viagens disponíveis, porém ainda poderá conter minutos restantes. Desta forma, é imprescindível verificar se ambos os indicadores estão a zeros no momento

da validação. É importante observar que este tipo de fraude poderá ser bastante difícil de identificar nos dados que serão analisados por esta regra. Isto deve-se ao facto de que a Horários do Funchal tem implementado mecanismos num nível anterior, os quais visam bloquear as validações de títulos que não possuem viagens antes da validação e não possuem minutos de viagem restantes no momento da validação.

- A quarta regra analisa se não houve consumo da viagem em entradas. Esta regra realiza uma verificação nos bilhetes pré-comprados para confirmar se o número de viagens antes e depois da validação permanece o mesmo quando se trata de uma entrada (TipoEvento = 1). Neste contexto, durante a entrada, a viagem é consumida, ao contrário de uma reentrada, onde a viagem não é descontada. Se durante uma entrada a viagem não for consumida, ou seja, se o valor do contador antes da entrada (*CounterValueBefore*) for igual ao valor do contador após a entrada (*CounterValueAfter*), a classificação será "Erro do Sistema". Caso contrário, a classificação será "Legítimo". É importante notar que, neste cenário, estamos lidando com um erro do sistema, e não necessariamente com um caso de fraude. Geralmente, estes erros são decorrentes de falhas na programação do sistema, em vez de serem resultado de atividades fraudulentas deliberadas. Embora casos como estes possam surgir, como observado nos dados analisados pela regra, é considerado um problema intrínseco ao funcionamento do sistema. Nestes casos, é razoável supor que a Horários do Funchal não tenha implementado um sistema anterior específico para detetar e resolver estas situações, já que a solução reside principalmente na correção do erro de programação.
- A quinta regra examina a validação de passes ou bilhetes de criança em intervalos de tempo incomuns, como durante a madrugada. Esta regra estabelece um intervalo de tempo com um limite inferior de "00:00" e um limite superior de "06:30". Posteriormente, a hora da validação extraída e guardada anteriormente no campo "ValidacaoHora" é avaliada para determinar se está dentro deste intervalo. Em seguida, a regra verifica se o registo de validação corresponde a um bilhete pré-comprado para crianças e se a hora extraída está dentro do intervalo definido. Se esta condição for verificada, a validação é considerada como suspeita de fraude, caso contrário, a regra classifica-a como legítima. Nesta regra, abordamos situações que possam parecer suspeitas ou incomuns, não necessariamente indicativas de fraude imediata. Isso deve-se ao facto de que, por exemplo, uma criança pode eventualmente viajar de autocarro durante a madrugada, acompanhada pelos pais, por mera coincidência, embora seja uma ocorrência pouco comum. Portanto, se houver uma ocorrência desse tipo num mês para o mesmo título, não é suficiente para concluir imediatamente que se trata de uma fraude. No entanto, se houver múltiplas validações desse mesmo título para crianças durante um intervalo de tempo específico, pode indicar uma possível fraude, dado que é altamente improvável que uma criança utilize regularmente o transporte público durante a madrugada.
- A sexta regra inspeciona bilhetes pré-comprados de múltiplas viagens por dias, verificando tanto os minutos restantes como possíveis excessos de minutos permitidos. Esta regra inicia-se com a criação de uma abordagem com *switch*/*case* que associa os diferentes tipos de bilhetes pré-comprados com

a sua respetiva validade em dias. A regra avalia se uma validação é considerada fraudulenta, comparando o número de dias e o tempo de viagem restante, ou seja, se o título estiver na lista, o número de dias coincidir e o tempo de viagem restante for zero, a validação é considerada como fraude. Adicionalmente, é realizada uma verificação para determinar se o tempo restante em minutos excede o permitido para o tipo específico de bilhete pré-comprado, por exemplo, caso um tenha validade para 5 dias, mas o tempo restante em minutos é 1 milhão. Se essa condição for verdadeira, a validação é classificada como "Fraude / Erro do Sistema". Além disso, a regra verifica se o número de dias do bilhete difere do número de dias lido na validação, por exemplo, caso um bilhete pré-comprado tenha validade para 3 dias, mas a validação indica que o número de dias é 10. Caso esta condição seja satisfeita, a transação é classificada igualmente como "Fraude / Erro do Sistema". Se nenhuma das verificações anteriores for atendida ou se o título não estiver na lista, a regra atribui uma classificação de legitimidade à validação. Na primeira verificação, abordamos situações de fraude, contudo, a ocorrência desses casos nos dados analisados por esta regra é improvável, porque o validador instalado na viatura vai indicar que a validação foi recusada pois não têm tempo restante de viagem. Quanto à segunda e à terceira verificação, que abordam casos interpretáveis como fraude ou erro do sistema, são pouco prováveis que ocorram essas situações, pois geralmente é atribuído automaticamente o número correto de minutos durante a compra e a criação do bilhete pré-comprado. Portanto, somente um erro no sistema ou manipulação intencional resultaria na criação desses casos.

- A sétima regra verifica se os passes de 1 mês e 12 messes possuem o número de meses adequado. Esta regra é estabelecida através da criação de um *switch*/*case* que associa os distintos passes com os seus períodos de validade, expressos em meses. O propósito primordial desta regra é assegurar que o passe validado apresente o número de meses adequado para a sua categoria correspondente, por exemplo, um passe mensal possui um número de meses igual a 12. Em essência, verifica se o passe validado está registado na lista, no qual guarda numa variável o número de meses correto. Em seguida, realiza uma comparação para confirmar se o valor na variável coincide com o valor da coluna "NumMeses", onde qualquer disparidade, pode indicar uma possível fraude ou falha no sistema. Na ausência da satisfação desta condição ou se o passe não constar na lista, a regra classifica como legítimo. Dificilmente esta regra irá dar classificações como fraudes ou erro do sistema, pois tal como a regra anterior, quando o passe é criado ou renovado é atribuído automaticamente o seu prazo de validade. Desta forma, só falhas no sistema ou alteração ilegítima é que poderiam original estes casos.
- A oitava regra segue um princípio muito semelhante à regra anterior, porém, a sua aplicação é direcionada aos passes de 15 e 30 dias, para garantir que os números de dias possuem o número de dias correto. É escusado repetir o que já foi mencionado anteriormente, a única diferença relevante a ser destacada é que, nesta regra, ao invés da coluna "NumMeses", utiliza-se a coluna "NumDias".

- A nona regra deteta validações muito próximas para o mesmo NSCartao, considerando uma nova validação entre o momento da validação anterior aceite e o término do seu período de *cooldown* (5 minutos depois). Primeiramente, esta regra eliminará temporariamente todos os caracteres não numéricos das datas e horas, gerando assim um número inteiro para facilitar as comparações. Esta regra tem como objetivo verificar se a data e hora da validação situa-se entre a data e hora da validação anterior do mesmo cliente e o término do período de *cooldown*, além de confirmar se a anterior foi aceite (PosValid = 1). Se esta condição for atendida, surge uma suspeita de fraude. Caso a condição não seja satisfeita, a regra classifica a validação como legítima. Nesta regra, abordamos cenários suspeitos em que não podemos imediatamente determinar como fraude, porque é comum ocorrerem múltiplas validações muito próximas involuntariamente, como quando o validador lê o título duas vezes ou quando o passageiro valida o título sem querer duas vezes. Portanto, um número reduzido de validações quase simultâneas para o mesmo cliente não é suficiente para concluir imediatamente que se trata de uma fraude. Por outro lado, se houver um número significativo de validações quase simultâneas, isso pode indicar uma potencial fraude, uma vez que não é comum haver muitas validações num intervalo tão curto de tempo.
- A décima regra verifica se há mais do que uma validação para o mesmo NSCartao no mesmo autocarro e na mesma viagem. Primeiramente, esta regra eliminará temporariamente todos os caracteres não numéricos das datas e horas, gerando assim um número inteiro para facilitar as comparações. Posteriormente, todos os registos que tenham a data e hora de início da viagem ("DHIniViagem") inválido, que ocorre quando o motorista não inicia a viagem, são classificados como "Dados em Falta". Se "DHIniViagem" for inválido, "UltimoDHIniViagem" da validação seguinte do mesmo cliente também será inválido. No entanto, mesmo que posteriormente haja uma transação onde "DHIniViagem" seja válido, mas se "UltimoDHIniViagem" for inválido, torna-se impossível classificar, exceto a primeira validação do cliente. Isso ocorre porque não é possível comparar uma data válida com uma data inválida, impossibilitando qualquer conclusão e, dessa forma, o sistema atribuí a classificação "Inválido" nesse registo, tal como a Tabela 2 ilustra.

Data	NSCartao	DHIniViagem	UltimoDHIniViagem	Classificação
25/07/2024 09:25:00	1	0000-00-00 00:00:00	0000-00-00 00:00:00	Dados em Falta
25/07/2024 09:25:01	2	25/07/2024 09:25:00	0000-00-00 00:00:00	Legítimo
26/07/2024 11:32:41	1	26/07/2024 11:01:55	0000-00-00 00:00:00	Inválido
27/07/2024 13:45:44	3	27/07/2024 11:30:22	0000-00-00 00:00:00	Legítimo
27/07/2024 13:47:12	2	27/07/2024 11:30:22	25/07/2024 09:25:00	Legítimo

28/07/2024 16:08:09	3	0000-00-00 00:00:00	27/07/2024 11:30:22	Dados em Falta
28/07/2024 16:08:10	4	28/07/2024 16:08:01	0000-00-00 00:00:00	Legítimo
28/07/2024 23:55:10	3	28/07/2024 23:45:47	0000-00-00 00:00:00	Inválido

Tabela 1 - Dados em falta e classificações inválidas

Em seguida, verifica se a validação atual ocorre no mesmo veículo que a validação anterior para o mesmo NSCartao, além de confirmar se a data e hora de início da viagem do veículo em que o passe ou bilhete pré-comprado está sendo validado corresponde à validação anterior, e se esta foi aceite. Se esta condição for satisfeita, a regra classifica a validação como fraudulenta, caso contrário, classifica-a como legítima, como pode ser verificado na tabela 2.

Data	NSCartao	DHIniViagem	UltimoDHIniViagem	Veículo	UltimoVeículo	PosValid	UltimoPosValid	Classificação
25/07/2024 09:25:00	1	25/07/2024 09:25:00	0000-00-00 00:00:00	100	0	1	0	Legítimo
25/07/2024 09:25:01	2	25/07/2024 09:25:00	0000-00-00 00:00:00	100	0	1	0	Legítimo
26/07/2024 11:32:41	1	25/07/2024 09:25:00	25/07/2024 09:25:00	100	100	1	1	Fraude 10
27/07/2024 13:45:44	3	27/07/2024 11:30:22	0000-00-00 00:00:00	250	0	1	0	Legítimo
27/07/2024 13:47:12	2	27/07/2024 11:30:22	25/07/2024 09:25:00	250	100	1	1	Legítimo
27/07/2024 13:47:13	5	27/07/2024 11:30:22	0000-00-00 00:00:00	250	0	0	0	Legítimo
27/07/2024 13:47:14	5	27/07/2024 11:30:22	27/07/2024 11:30:22	250	250	1	0	Legítimo
28/07/2024 16:08:09	3	28/07/2024 16:08:01	27/07/2024 11:30:22	1001	250	1	1	Legítimo
28/07/2024 16:08:10	4	28/07/2024 16:08:01	0000-00-00 00:00:00	1001	0	1	0	Legítimo
28/07/2024 23:55:10	3	28/07/2024 16:08:01	28/07/2024 16:08:01	1001	1001	1	1	Fraude 10

Tabela 2 - Validações fraudulentas

Se um passageiro sair do autocarro na paragem errada por engano e desejar coltar a entrar imediatamente no mesmo veículo, ele será impedido de fazê-lo, pois isso iria contrariar esta regra. Neste contexto, o passageiro é obrigado a adquirir um bilhete de bordo conforme as políticas estabelecidas pela Horários do Funchal. Dessa forma, a ocorrência de fraudes deste tipo é pouco provável nos dados disponibilizados.

- A décima primeira regra verifica a validação de bilhetes pré-comprados e passes urbanos tanto em operadores interurbanos quanto na rede interurbana. Esta regra possui uma abordagem com *switch*/*case* que compreende uma lista de passes e bilhetes pré-comprados urbanos, os quais são destinados exclusivamente para uso em área urbanas, especialmente no Funchal. A regra é projetada para verificar a inclusão do passe ou bilhete pré-comprado validado nessa lista e determinar se o mesmo foi validado tanto num operador interurbano quanto pela rede de transportes interurbano. Caso esta condição não seja cumprida, ou se não estiver presente na lista, a validação é considerada legitima, no entanto, se a validação ocorrer no operador ou na rede mencionada, a mesma é categorizada como "Fraude / Erro do Sistema". É pouco provável que ocorram situações interpretáveis como fraudes ou erros do sistema dentro da análise desta regra. Isto deve-se ao facto de que durante o processo de compra, criação e renovação do título é atribuída automaticamente a rede apropriada. Adicionalmente, a validação também será recusada no momento que o passageiro coloque o seu título no validador.
- A décima segunda regra verifica se a tarifa correta foi aplicada na validação dos passes e bilhetes pré-comprados. Esta regra também utiliza uma estrutura de *switch*/*case* que inclui uma lista de tarifas distintas, nomeadamente tarifa 1, tarifa 2 e tarifa 3, bem como uma relação de passes e bilhetes pré-comprados, associados a cada tarifa específica. A regra busca o título na referida lista e, ao encontrá-lo, verifica se o tipo de tarifa corresponde com o da validação. Caso contrário, se houver uma diferença, a regra identifica esse cenário como uma Fraude / Erro do Sistema. Por outro lado, se o título não for encontrado na lista passa para a seguinte, se não estiver em nenhuma lista o caso é classificado como legítimo. São improváveis as situações que possam ser classificadas como fraude ou erro do sistema ao aplicar esta regra pelo mesmo motivo da regra anterior.
- A décima terceira regra verifica se os bilhetes ou passes validados estão atribuídos ao grupo correto. Esta regra igualmente opera com base num *switch*/*case*, onde contém uma lista de bilhetes para o Grupo 1 e uma lista de passes para o Grupo 2. Inicialmente, a regra verifica se o bilhete ou passe validado está presente na primeira, em caso negativo, verifica se está na segunda lista, se não for encontrado, é atribuído o valor padrão. Após localizar o bilhete ou passe, verifica se o grupo associado é o correto, comparando com o valor da coluna "GrupoTitulo". Se for identificado como diferente, a regra classifica a validação como uma Fraude / Erro do Sistema, caso contrário, ou se não for encontrado, é considerada legítima. É improvável que a análise desta regra resulte em interpretações de fraude ou falhas do sistema, a justificação da regra anterior ou da décima primeira regra aplica-se também aqui.

- **SEND CLASSIFICATION TO DB** - Este step implica a execução de uma *query* que irá obter todas as classificações e as inserirá na base de dados, atribuindo-as à sua respetiva coluna de acordo com o número do registo, garantindo assim a associação precisa de cada registo na tabela com a classificação correspondente.

4.4. Testes

A partir deste ponto, é essencial compreender como as classificações fornecidas pelo sistema de deteção de fraudes são atribuídas. Cada classificação indica a regra específica que foi satisfeita, permitindo determinar se a validação é legítima ou não.

Por exemplo:

- Uma classificação "Fraude 1" indica que a primeira regra identificou a validação como fraude.
- "Erro do Sistema 11" significa que a décima primeira regra identificou um erro na validação.
- "Fraude / Erro do Sistema 7.1" indica que a primeira condição da sétima regra classificou a validação como fraude, que também poderá ser erro do sistema, dependendo do cenário. Da mesma forma, "Fraude / Erro do Sistema 7.2" significa que a segunda condição da sétima regra fez essa classificação.
- A classificação "Inválido" é atribuída quando não é possível chegar a uma conclusão específica.
- "Erro do Sistema 13.1" indica que a primeira condição da décima terceira regra identificou um erro na validação.
- "Fraude / Erro do Sistema 12" significa que a décima segunda regra classificou a transação como podendo ser uma fraude ou um erro do sistema, dependendo do cenário.
- "Suspeita de Fraude 10" sugere que a décima regra identificou um potencial caso de fraude que requer atenção adicional.

As demais classificações seguem o mesmo padrão de nomenclatura e interpretação descrito acima.

Os testes seguem uma estrutura bem definida, que se inicia com a inserção ou modificação de dados, seguida da execução do sistema para aplicar as regras, e termina com *queries* de verificação, que confirmam se as classificações atribuídas correspondem aos cenários simulados.

Todas as regras e classificações serão submetidas a testes individuais, ou seja, serão avaliadas uma a uma. Por outras palavras, é essencial desativar temporariamente as demais regras para assegurar que os resultados obtidos reflitam exclusivamente as classificações geradas pela regra que está sendo testada. Esta abordagem permite uma avaliação mais precisa do seu desempenho, e assim determinar mais facilmente se as validações foram corretamente classificadas. Consequentemente, cada regra e sub-regra será aplicada individualmente, e após essa aplicação, novas *queries* serão realizadas. De modo geral, uma *query* será feita para visualizar os casos legítimos e outras para visualizar os casos não legítimos. Espera-se que, em uma delas, todas as validações sejam todas legítimas, enquanto nas outras, todas sejam não legítimas. Esta abordagem permite uma verificação detalhada da eficácia das regras e sub-regras, assegurando que elas desempenham as suas funções corretamente. Estes processos de testes têm como objetivo garantir a precisão e a confiabilidade do sistema de deteção de fraudes baseado em regras.

Vale a pena ainda referir que os testes seguirão uma metodologia na qual situações não legítimas poderão ser criadas deliberadamente por meio de instruções diretamente na base de dados, com o objetivo de verificar se a regra ou sub-regra classifica corretamente cada caso.

No entanto, tendo em conta a repetição de procedimentos ao longo das 13 regras definidas, optou-se por manter neste capítulo apenas as três primeiras regras com abordagens distintas:

- A primeira regra, que envolve a inserção de novos dados;
- A segunda regra, baseada na modificação de dados;
- A sexta regra, que apresenta múltiplas condições e maior complexidade.

As regras restantes (3.ª a 5.ª e 7.ª a 13.ª) são documentadas em detalhe no Anexo A – Testes das Regras 3 a 13, de forma a não sobrecarregar a leitura principal mantendo, no entanto, a completude técnica do trabalho.

4.4.1. Primeira regra

Para avaliar a aplicação da primeira regra, empregou-se a *query* descrita abaixo para clonar certas linhas na tabela. A funcionalidade desta *query* reside na seleção aleatória de uma linha entre os identificadores (coluna "Linha") 1035005530 e 1035955828 e a sua posterior inserção na tabela, onde o processo é interrompido após a duplicação de 1000 linhas, uma quantidade que se considera adequada para obter uma precisão interessante sobre o desempenho da regra. É importante referir que o número específico de duplicações pode ser ajustado conforme pretendido, visando garantir uma análise completa do comportamento da regra em questão. A clonagem de determinadas linhas da tabela tem como finalidade criar um cenário no qual ocorrem duas ou mais validações simultâneas. Com este cenário estabelecido e considerado a condição da regra em questão, espera-se que ela classifique estes casos como "Fraude 1".

Um aspeto a ser considerado é a possibilidade da variação no intervalo das linhas selecionadas, influenciada pelo tamanho atual da tabela. Durante o desenvolvimento do sistema no Pentaho, um *step* foi implementado para limitar o número de linhas que entram na transformação e, consequentemente, na base de dados, o que resultaria em prolongados tempos de processamento durante as fases de desenvolvimento e teste. Por outras palavras, cada alteração na transformação e os testes das regras poderiam exigir aproximadamente 1 hora, o que causaria demoras no seu desenvolvimento. O problema deste *step* surge na sua precisão e estabilidade, ou seja, em algumas ocasiões o limite estabelecido não é alcançado, resultando num bloqueio bem antes do estipulado. Além disso, observou-se que o número de linhas pode varia entre execuções, como bloquear às 8000 linhas numa execução e às 7000 linhas noutras, apesar do limite ser de 10000 linhas. Este problema é inerente ao próprio funcionamento do Pentaho e do seu *step*, e que, até ao momento, não foi possível identificar nenhuma solução para resolver esta questão. No entanto, é de referir que este *step* está sendo utilizado apenas durante as fases de desenvolvimento, para avaliação das regras implementadas e em demonstrações. Quando for necessário realizar testes mais abrangentes, essa etapa será removida ou terá o seu limite aumentado. Em ambientes de produção, por sua vez, será definitivamente eliminado.

O código 1 em MySQL insere 1.000 registos aleatórios selecionados da tabela "dataoutput" de volta na mesma tabela. Resumidamente ele faz o seguinte:

- **INSERT INTO dataoutput (...)**: Define a tabela de destino ("dataoutput") e as colunas onde os dados serão inseridos.
- **SELECT ... FROM dataoutput**: A operação SELECT extrai os dados da mesma tabela ("dataoutput"), em vez de uma nova fonte de dados, e define quais colunas vão compor cada novo registo.
    - **(SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha e (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo**: Para garantir que os campos Linha e Codigo dos novos registos sejam únicos, o código obtém o valor máximo atual dessas colunas na tabela "dataoutput" e adiciona 1.
- **WHERE Linha BETWEEN 1035005530 AND 1035955828**: Limita a seleção de registos a uma faixa específica de valores de Linha.
- **ORDER BY RAND() LIMIT 1000**: Ordena os resultados aleatoriamente e limita a quantidade a 1.000 registos. Assim, cada execução desta consulta insere uma amostra aleatória de registos dessa faixa na tabela.

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035955828
```

```
ORDER BY RAND()
LIMIT 1000;
```

Código 1 - Inserção de novos dados (1ª Regra)

Já no código 2, a primeira *query* tem como objetivo selecionar registos onde o "NSCartao" (número de cartão) aparece mais de uma vez (**HAVING COUNT(*) > 1**) e onde a data de "DataHora" é igual à "UltimaValidacao". Em primeiro lugar, uma subconsulta (**INNER JOIN**) agrupa os cartões ("NSCartao") e retém apenas aqueles que aparecem em mais do que um registo. Em seguida, o **INNER JOIN** vincula esses cartões repetidos com o conjunto de dados original ("dataoutput"). Por fim, o filtro **WHERE d.DataHora = d.UltimaValidacao** garante que sejam exibidos apenas os registos em que o "DataHora" coincide com "UltimaValidacao".

A segunda *query* tem o objetivo semelhante à primeira consulta, mas agora busca os registos onde a "DataHora" é diferente de "UltimaValidacao". A estrutura é idêntica à da primeira consulta, mas o filtro **WHERE d.DataHora != d.UltimaValidacao** agora foca nos casos onde "DataHora" e "UltimaValidacao" são distintos.

```
SELECT d.DataHora, d.UltimaValidacao, d.NSCartao, d.Classificacao
FROM dataoutput d
INNER JOIN (
   SELECT NSCartao
   FROM dataoutput d
   GROUP BY NSCartao
   HAVING COUNT(*) > 1
) b ON d.NSCartao = b.NSCartao where d.DataHora = d.UltimaValidacao;

SELECT d.DataHora, d.UltimaValidacao, d.NSCartao, d.Classificacao
FROM dataoutput d
INNER JOIN (
   SELECT NSCartao
   FROM dataoutput d
   GROUP BY NSCartao
   HAVING COUNT(*) > 1
) b ON d.NSCartao = b.NSCartao where d.DataHora != d.UltimaValidacao;
```

Código 2 - Avaliação das classificações (1ª Regra)

Ao executar as duas *queries*, analisamos todas as linhas resultantes e observamos que, em ambas, a classificação foi constantemente "Fraude 1" na primeira e "Legítimo" na segunda, num total de 2000 registos em cada caso, proporcionando assim um teste robusto. Com base nestes resultados, pode-se concluir que a regra está classificando dos dados de forma precisa.

Durante os testes iniciais, foi identificado um problema na regra, onde mesmo quando os campos "DataHora" e "UltimaValidacao" eram iguais, ela classificava erradamente os dados como legítimos em vez de fraudulentos. Esta inconsistência indicava que a condição da regra não estava sendo satisfeita adequadamente, resultando numa interpretação incorreta dos dados. Embora ambos os campos fossem do mesmo tipo de dado – *string* – e os seus valores fossem iguais, a condição de comparação não estava sendo atendida. Para resolver este problema, foi adotada a estratégia que pode ser vista na secção 4.5. Discussão dos testes.

4.4.2.Segunda regra

Para avaliar a eficácia da segunda regra, foi empregue a seguinte *query* para modificar aleatoriamente alguns valores do campo "TipoEvento" para "3", indicando que o bilhete de bordo ou o bilhete de bordo para adultos validado representa uma reentrada. Conforme discutido anteriormente, a regra tem de ser capaz de classificar esses casos como fraudulentos, uma vez que os bilhetes de bordo são destinados a uma única utilização, enquanto os outros casos devem ser considerados legítimos.

O código 3 em MySQL atualiza o campo TipoEvento na tabela "dataoutput" com base em condições específicas:

- **UPDATE dataoutput**: Define a tabela "dataoutput" como o alvo da atualização.
- **SET TipoEvento = CASE WHEN (Linha % 3) = 0 THEN '3' ELSE '1' END**:
    - Esta expressão usa a instrução **CASE** para condicionalmente definir o valor de "TipoEvento":
        - **WHEN (Linha % 3) = 0 THEN '3'**: Quando o valor da coluna "Linha" é divisível por 3 (o que significa que o resto da divisão Linha % 3 é 0), "TipoEvento" será definido como "3".
        - **ELSE '1'**: Para todos os outros valores de "Linha" (ou seja, aqueles não divisíveis por 3), "TipoEvento" será definido como "1".
- **WHERE Titulo = 4880 OR Titulo = 4817**: Esta cláusula limita a atualização aos registros onde "Titulo" é igual a "4880" ou "4817".

```
UPDATE dataoutput SET TipoEvento = CASE WHEN (Linha % 3) = 0 THEN '3' ELSE '1' end where Titulo = 4880 or Titulo = 4817;
```

Código 3 - Modificação dos dados existentes (2ª Regra)

É importante referir que antes da realização da classificação dos dados, foi necessário realizar ajustes nas colunas "Titulo" e "descrição" de determinadas linhas. Estas modificações foram aplicadas em registos que inicialmente eram referentes a validações de bilhetes de bordo, transformando-os em validações de bilhetes de bordo para adultos. Esta intervenção foi necessária devido à ausência de registos desse tipo de bilhetes no conjunto de dados original. Desta forma, estas alterações têm o objetivo de testar a capacidade da regra de classificação em lidar adequadamente com validações de bilhetes de bordo para adultos.

Relativamente ao código 4, a primeira *query* tem como objetivo extrair combinações únicas de "TipoEvento", "Descricao" e "Classificacao" onde o "Titulo" seja igual a "4880" ou "4817" e o "TipoEvento" seja "3".

- **SELECT DISTINCT**: A cláusula DISTINCT garante que apenas linhas únicas (sem repetições) sejam retornadas.
- **WHERE (Titulo = 4880 OR Titulo = 4817) AND TipoEvento = 3**: Filtra os registos onde "Titulo" é "4880" ou "4817" e "TipoEvento" é "3".

A segunda *query* funciona de maneira similar à primeira consulta, mas agora extrai registos onde o "TipoEvento" é "1". Por outras palavras, a estrutura é idêntica, mas o filtro **WHERE** agora especifica **TipoEvento = 1**.

```
select distinct TipoEvento, Descricao, Classificacao from dataoutput d where (Titulo = 4880 or Titulo = 4817) and TipoEvento = 3;

select distinct TipoEvento, Descricao, Classificacao from dataoutput d where (Titulo = 4880 or Titulo = 4817) and TipoEvento = 1;
```

Código 4 - Avaliação das classificações (2ª Regra)

Ao executar estas duas *queries*, foram obtidos os resultados "3 | Bilhete de Bordo | Fraude 2" e "3 | B.Bordo Adulto | Fraude 2" e "1 | Bilhete de Bordo | Legítimo" e "1 | B.Bordo Adulto | Legítimo", respetivamente. Este resultado sugere que, dentro do conjunto de dados analisado, quando a coluna "TipoEvento" contém o valor "3" e a coluna "Titulo" contém os valores "4880" ou "4817", que correspondem aos identificadores para o bilhete de bordo e bilhete de bordo para adultos, respetivamente, a validação é constantemente classificada como "Fraude 2". Por outro lado, quando a coluna "TipoEvento" possui o valor "1" para os mesmos títulos, a classificação é sempre "Legítima", conforme estipulado pela segunda regra.

4.4.3.Sexta regra

No âmbito da avaliação da sexta regra, propõe-se uma modificação específica na coluna "TempoViagemRestanteMins", com o propósito de alterar o seu valor para zero nas validações de bilhetes pré-comprados para multiviagens.

Além disso, é proposta outra modificação na coluna "TempoViagemRestanteMins". Esta alteração tem como objetivo ajustar o valor desta coluna para que seja superior ao valor presente na coluna "NumDias", expresso em minutos, nas validações dos bilhetes já mencionados, em alguns registos já existentes na tabela da base de dados. Por exemplo, se a coluna "NumDias" possuir o valor "1", então o valor na coluna "TempoViagemRestanteMins" será modificado para ser superior a 1440 minutos. Nesta *query* em concreto foi escolhido um valor bastante grande para ter a garantia que esse valor é sempre superior ao valor da coluna "NumDias".

Por fim, podemos realizar mais uma modificação nos valores da coluna "NumDias" da tabela. Esta alteração consiste em atribuir valores irrealistas, com a finalidade de haver certezas de que estes valores nunca coincidam com o número correto de dias para os títulos em questão. O objetivo é criar uma configuração na qual os valores não correspondem ao número correto de dias para os respetivos títulos.

Estas abordagens buscam classificar tais ocorrências como “Fraude 6”, “Fraude / Erro do Sistema 6.1” e “Fraude / Erro do Sistema 6.2”, respetivamente, conforme os critérios estabelecidos pela regra em análise.

O código 5 em MySQL consiste em três instruções UPDATE que alteram os valores de diferentes colunas na tabela “dataoutput”. Essas instruções têm como alvos registos que possuem títulos específicos.

A primeira instrução tem como objetivo atualizar o valor da coluna “TempoViagemRestanteMins” para 0 nos registos que têm um dos títulos especificados.

- A cláusula **SET** define que “TempoViagemRestanteMins” será alterado para “0”.
- A cláusula **WHERE** especifica que somente os registos com os títulos “5076”, “4858”, “4864”, “4870”, “3744” ou “3804” serão afetados.
- O uso do **LIMIT 100** significa que a atualização se aplicará apenas aos primeiros 100 registos que atendem à condição.

A segunda instrução tem como objetivo atualizar o valor da coluna “TempoViagemRestanteMins” para 1.000.000 nos registos que têm um dos títulos especificados.

- Similar à primeira instrução, mas aqui o valor de “TempoViagemRestanteMins” é definido como “1.000.000”.
- Os mesmos títulos são utilizados na cláusula **WHERE**.
- O **LIMIT 100** novamente limita a atualização aos primeiros 100 registos que correspondem à condição.

A terceira instrução tem como objetivo atualizar o valor da coluna “NumDias” para 1.000.000 nos registos que têm um dos títulos especificados.

- Aqui, a coluna “NumDias” é alterada para “1.000.000”.
- A condição **WHERE** é a mesma, utilizando os mesmos títulos.
- O **LIMIT 100** novamente garante que apenas os primeiros 100 registos que atendem ao critério serão atualizados.

```
update dataoutput set TempoViagemRestanteMins = 0 where Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo = 3744 or Titulo = 3804 limit 100;

update dataoutput set TempoViagemRestanteMins = 1000000 where Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo = 3744 or Titulo = 3804 limit 100;

update dataoutput set NumDias = 1000000 where Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo = 3744 or Titulo = 3804 limit 100;
```

Código 5 - Modificação dos dados existentes (6ª Regra)

No entanto, a avaliação apresentará uma abordagem distinta em relação às avaliações anteriores, ou seja, será dividida em três partes, em virtude de possuir três condições. Optou-se por esta abordagem pois não é recomendável separá-las em regras distintas, uma vez que elas analisam validações dos mesmos títulos e duas delas evocam a mesma função. Portanto, para evitar redundâncias, é mais adequado manter as condições juntas, mas para garantir uma maior precisão, cada uma será avaliada individualmente.

Primeira condição

No código 6 em MySQL a primeira consulta tem como objetivo selecionar os registos que têm "TempoViagemRestanteMins" igual a 0 e que possuem um dos títulos especificados.

- A consulta utiliza a cláusula **SELECT** para buscar as colunas "Descricao", "Titulo", "TempoViagemRestanteMins" e "Classificacao".
- A cláusula **WHERE** contém duas condições:
    - **TempoViagemRestanteMins = 0**: Isto filtra os registos para incluir apenas aqueles que não têm tempo de viagem restante.
    - **(Titulo = 5076 OR Titulo = 4858 OR Titulo = 4864 OR Titulo = 4870 OR Titulo = 3744 OR Titulo = 3804)**: Esta condição garante que apenas os registos com os títulos especificados sejam incluídos no resultado.

A segunda consulta tem como objetivo selecionar registos que têm "TempoViagemRestanteMins" diferente de 0 e que possuem um dos títulos especificados.

- A consulta segue a mesma estrutura da primeira, mas com uma condição diferente na cláusula **WHERE**:
    - **TempoViagemRestanteMins != 0**: Isto filtra os registos para incluir apenas aqueles que têm um tempo de viagem restante maior que 0.
    - A segunda parte da condição (os títulos) permanece a mesma, garantindo que apenas os registos com os títulos especificados sejam considerados.

```
select Descricao, Titulo, TempoViagemRestanteMins, Classificacao from dataoutput where
TempoViagemRestanteMins = 0 and (Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo =
3744 or Titulo = 3804);

select Descricao, Titulo, TempoViagemRestanteMins, Classificacao from dataoutput where
TempoViagemRestanteMins != 0 and (Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo
= 3744 or Titulo = 3804);
```

Código 6 - Avaliação das classificações (6ª Regra – 1ª Condição)

Após a execução das duas *queries*, observou-se que todas as validações foram categorizadas como "Fraude 6.1" na primeira e como "Legítimo" na segunda. Vale a pena referir que foram retornadas 100 registos e 42 registos, respetivamente. Com este cenário, é possível afirmar que a regra classificou as validações corretamente.

Segunda condição

No código 7 em MySQL a primeira *query* tem como objetivo selecionar registos onde o tempo de viagem restante em minutos é maior que a conversão dos dias em minutos e filtrar por títulos específicos.

- A consulta seleciona as colunas "Descricao", "Titulo", "TempoViagemRestanteMins", "NumDias", e também calcula "NumDiasEmMinutos" (o total de minutos correspondente ao número de dias multiplicando "NumDias" por 1440, já que 1 dia = 1440 minutos).
- A cláusula **WHERE** contém duas condições:
    - **TempoViagemRestanteMins > (NumDias * 1440)**: Isto filtra os registos para incluir apenas aqueles onde o tempo restante da viagem em minutos é maior que o total de minutos calculados a partir de "NumDias".
    - **(Titulo = 5076 OR Titulo = 4858 OR Titulo = 4864 OR Titulo = 4870 OR Titulo = 3744 OR Titulo = 3804)**: Esta condição garante que apenas os registros com os títulos especificados sejam incluídos no resultado.

A segunda *query* tem como objetivo selecionar os registos onde o tempo de viagem restante em minutos é menor ou igual à conversão dos dias em minutos, filtrando também por determinados títulos.

- Assim como na primeira consulta, esta também seleciona as mesmas colunas, incluindo o cálculo de "NumDiasEmMinutos".
- A cláusula **WHERE** agora usa uma condição diferente:
    - **TempoViagemRestanteMins <= (NumDias * 1440)**: Esta condição filtra os registos para incluir aqueles em que o tempo restante da viagem em minutos é menor ou igual ao total de minutos calculados a partir de "NumDias".
    - A condição para os títulos permanece a mesma, assegurando que apenas os registos com os títulos especificados sejam considerados.

```
SELECT Descricao, Titulo, TempoViagemRestanteMins, NumDias, NumDias * 1440 AS
NumDiasEmMinutos, Classificacao
FROM dataoutput
WHERE TempoViagemRestanteMins > (NumDias * 1440)
  AND (Titulo = 5076 OR Titulo = 4858 OR Titulo = 4864 OR Titulo = 4870 OR Titulo = 3744 OR Titulo =
3804);

SELECT Descricao, Titulo, TempoViagemRestanteMins, NumDias, NumDias * 1440 AS NumDiasEmMinutos,
Classificacao
FROM dataoutput
WHERE TempoViagemRestanteMins <= (NumDias * 1440)
```

```
 AND (Titulo = 5076 OR Titulo = 4858 OR Titulo = 4864 OR Titulo = 4870 OR Titulo = 3744 OR Titulo = 3804);
```

Código 7 - Avaliação das classificações (6ª Regra – 2ª Condição)

A execução de ambas as *queries* revelou que todas as 100 validações retornadas pela primeira foram classificadas como "Fraude / Erro do Sistema 6.2" e todas as 149 validações retornadas pela segunda *query* foram classificadas como "Legítimo". Com base nestes resultados, podemos afirmar que a regra classificou as validações como desejado.

Terceira condição

No código 8 em MySQL o primeiro bloco tem como objetivo selecionar os registos onde o número de dias é maior que 7 e que correspondem a títulos específicos.

- A consulta seleciona as colunas "Descricao", "Titulo", "NumDias", e "Classificacao" da tabela "dataoutput".
- A cláusula **WHERE** contém duas condições:
    - **NumDias > 7**: Isto filtra os registos para incluir apenas aqueles onde a quantidade de dias é maior que 7.
    - **(Titulo = 5076 OR Titulo = 4858 OR Titulo = 4864 OR Titulo = 4870 OR Titulo = 3744 OR Titulo = 3804)**: Esta condição garante que somente os registos que possuem os títulos especificados sejam incluídos no resultado.

A segundo bloco tem como objetivo selecionar registos onde o número de dias é menor ou igual a 7 e que correspondem a determinados títulos.

- Assim como na primeira consulta, esta também seleciona as colunas "Descricao", "Titulo", "NumDias" e "Classificacao".
- No entanto, a cláusula **WHERE** utiliza uma condição diferente:
    - **NumDias <= 7**: Esta condição filtra os registos para incluir aqueles em que a quantidade de dias é menor ou igual a 7.
    - A condição para os títulos permanece a mesma, assegurando que apenas os registos com os títulos especificados sejam considerados.

```
select Descricao, Titulo, NumDias, Classificacao from dataoutput where NumDias > 7 and (Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo = 3744 or Titulo = 3804);

select Descricao, Titulo, NumDias, Classificacao from dataoutput where NumDias <= 7 and (Titulo = 5076 or Titulo = 4858 or Titulo = 4864 or Titulo = 4870 or Titulo = 3744 or Titulo = 3804);
```

Código 8 - Avaliação das classificações (6ª Regra – 3ª Condição)

Após a execução das consultas foi verificado que todas as validações receberam uma classificação de “Fraude / Erro do Sistema 6.3” na primeira e “Legítimo” na segunda. É de referir ainda que a primeira retornou um total de 100 registos, enquanto a segunda retornou 42. Com base nisto, podemos afirmar que a terceira condição da sexta regra está operando conforme o esperado.

4.5. Discussão dos testes

Inicialmente, optamos por utilizar constantes com o objetivo de aumentar a clareza do código, facilitando a sua compreensão e manutenção ao longo do tempo. Essa abordagem também iria promover um desenvolvimento mais consistente e robusto do sistema, reduzindo a probabilidade de erros. No entanto, durante os testes realizados, observamos que os tempos de classificação dos dados eram excessivamente altos devido ao uso frequente do método “parseInt()” em quase todas as constantes. Esse uso excessivo ocorre porque, por padrão, a maioria dos valores atribuídos em JavaScript são interpretados como decimais, mesmo quando são inteiros, consequentemente, essa conversão resulta em *bottlenecks* no código, causando uma perda de desempenho. Isso ocorre porque o código precisa constantemente converter ou forçar entre tipos de dados diferentes ou não, o que exige um processamento adicional e, portanto, tem um impacto negativo na eficiência do código. Desta forma, optou-se por eliminar todas as constantes, *steps* relacionados e utilizar diretamente os valores no código, tornando-o mais leve e mais rápido, o que é crucial para o processo de classificação dos dados. Para manter a manutenção do código simples, foi identificado o que cada valor representa na forma de comentário.

Outra dificuldade enfrentada foi na falha na comparação entre valores inteiros, mesmo sendo iguais, onde todos os dados estavam a ser classificados como legítimos, independentemente das circunstâncias. A documentação do Pentaho sugere que “se tiver enfrentando problemas ao usar == ou *switch* / *case* em que é sabido que são atribuídos como inteiro, então pode-se igualmente utilizar os seguintes métodos: parseInt(num) == parseInt(num2)” [32]. Desta forma, seguindo esta recomendação e minimizar a utilização do método “parseInt()”, a estrutura “switch / case” foi empregada na maior parte das situações, o que acabou por resolver a dificuldade, permitindo a correta classificação das regras. Esta abordagem não apenas também ajudou a reduzir a necessidade do método “parseInt()”, uma vez que a estrutura “switch / case” só exige valores literais nos “case”, mas também contribuiu para um código mais limpo e eficiente. Nas situações em que foi necessário utilizar os operadores lógicos “&&” e “||” e os operadores de comparação “==”, “>”, “<”, “>=”, “<=” e “!=”, empregou-se a estrutura “if / else” ou “if / if else / else”, no entanto, esta não elimina totalmente a necessidade do “parseInt()”, conforme a situação descrita a seguir. Em resumo, o método aqui em causa para converter as variáveis para inteiros é utilizado apenas para comparar duas variáveis; caso contrário, a classificação seria incorreta. Por exemplo, ao comparar variáveis com valores literais, como em “Titulo == 4817”, não é necessário usar o método “parseInt()”. Contudo, ao comparar duas variáveis, como em “parseInt(Dias) == parseInt(NumDias), o uso do mesmo é necessário.

Outra dificuldade encarada foi a falha na comparação de *strings*. Inicialmente, utilizou-se o operador “==” para comparar *strings*, mais isso resultou numa classificação incorreta, onde todas as validações estavam sendo atribuídas como legítimas, independentemente das circunstâncias. Posteriormente, ao usar o mesmo operador em conjunto com

o método “toString()” , as classificações passaram a ficar corretas. No entanto, a documentação do Pentaho recomenda evitar o uso do operador aqui em questão para comparar *strings*, sugerindo em vez disso o uso de “string.equals(otherString)” [32]. Após adotar essa recomendação e remover o uso dos métodos implementados, as classificações mantiveram-se corretas. Embora possa parecer insignificante, o “toString()”, assim como o “parseInt()”, pode ser um *bottleneck* de desempenho quando usado em excesso. Desta forma, a remoção desse método resultou num código mais eficiente. O “toString” é utilizado ao fazer comparações com operadores de comparação em duas variáveis do tipo *string*. Isso deve-se ao facto de que a abordagem recomendada anteriormente só é adequada para verificar se duas variáveis do tipo *string* são iguais.

Além disso, enfrentou-se outro desafio relacionado com um excerto do código da sexta regra, que envolvia a manipulação da coluna “DataHora”. O processo envolvia a extração apenas da parte referente à hora no formato HH:MM para realizar comparações com um intervalo predefinido. Contudo, essa abordagem resultava em classificações incorretas. Para resolver essa questão, implementou-se originalmente um *step* adicional antes da etapa “DATA CLASSIFICATION” que extraía a hora e o minuto do campo “DataHora” e armazenava num novo campo denominado “ValidacaoHora”, para preservar o valor original. Posteriormente, devido aos problemas de desempenho na transformação de classificação dos dados, migrou-se essa implementação para a transformação de limpeza dos dados no *step* “REMOVE SECONDS AND MILLISECONDS”, isso não só aumentou o desempenho, mas também passou a se concentrar apenas na classificação dos dados, sem tarefas adicionais que pudessem atrasar o processo.

É de salientar que, apesar de ter sido consideradas inicialmente, algumas regras não puderam ser implementadas. Especificamente, a regra que verifica cada validação de bilhete deve estar vinculada a uma venda efetiva não pôde ser aplicada devido à ausência de dados pertinentes no conjunto fornecido. Não só não há informações disponíveis para calcular estes dados a partir dos existentes, como também não há acesso ao sistema ou aos dados de vendas. Além disso, não foi possível criar a regra que deteta validações de bilhetes muitos distantes num curto período de tempo. Este é um resultado direto da inexistência de dados relevantes no conjunto fornecido, e não há meios disponíveis para derivar tais informações de dados existentes.

4.6. Reflexão dos testes

Conforme observado, todas as regras estão a classificar corretamente as validações, apesar dos contratempos encontrados, os quais foram resolvidos. Elas estão prontas para serem submetidas a um teste final, visando classificar todas as validações de forma integrada.

É importante salientar que este teste final será conduzido em subconjuntos de dados diários, com o propósito de avaliar o tempo de classificação e verificar se há variações significativas entre os dias ou se mantém constante ao longo do tempo. É esperado que eventuais discrepâncias no tempo de classificação sejam observadas, especialmente se um determinado dia apresentar um volume maior de validações em relação aos demais.

4.7. Teste final

Inicialmente, o plano para o teste final era dividi-lo em duas partes. Na primeira parte, pretendíamos utilizar o conjunto completo de dados fornecidos e segmentar em 12 partes distintas, correspondentes a cada mês do ano, no entanto, isso mostrou-se ser inviável. Testar apenas um mês levaria mais de dois dias completos sem interrupção, e estamos falando de 12 conjuntos.

Buscou-se aumentar ao máximo o desempenho do servidor MySQL para aproveitar melhor os recursos do computador onde ele está hospedado. Observou-se uma melhoria no desempenho das transformações e do próprio servidor; no entanto, os tempos de execução continuaram altos. É relevante deixar a nota de que o computador que possui o servidor já apresenta algumas limitações a nível de recursos devido à sua idade. Mesmo assim, foram mantidas as configurações para garantir o máximo desempenho recomendado do servidor. Assim sendo, tornou-se necessário buscar outras soluções que proporcionassem tempos de execução mais razoáveis, considerando as limitações.

Optou-se, então, por realizar os testes semanalmente para verificar se o processo seria mais rápido. De facto, houve uma melhoria, mas a estimativa de conclusão do teste ainda era muito alta, próximo a um dia completo sem interrupção. Desta forma, houve uma obrigação em testar dia a dia ao longo do ano de forma aleatória. Mesmo assim, cada teste diário demorou entre trinta minutos e duas horas e meia, dependendo do volume de validações desse dia.

Portanto, a segunda parte, que consistia em testar o conjunto completo de dados de uma só vez, tornou-se completamente inviável, pois exigiria um tempo considerável e contínuo para a conclusão do teste. No entanto, podemos estimar o tempo necessário para testar o conjunto completo de dados com base nas informações obtidas durante os testes diários.

Apesar destes contratempos, o objetivo primordial é aplicar as regras definidas em cada conjunto para avaliar não apenas o tempo de execução em cada segmento, mas também para identificar eventuais discrepâncias entre eles. É importante notar que tais disparidades podem surgir devido a diferenças na quantidade de validações realizadas em cada conjunto, sendo possível que um determinado dia apresente um volume maior em comparação aos demais.

Adicionalmente, podem ser aplicadas as *queries* de visualização que foram utilizadas nos testes anteriores. Este procedimento visa verificar se o sistema identificou corretamente os casos suspeitos, fraudulentos, erros do sistema, entre outros. Embora tenha sido constatado que as regras de classificação estão operando de forma precisa, é prudente realizar esta verificação adicional para assegurar a exatidão das classificações.

5. Resultados

Aqui são apresentados e analisados os resultados dos testes efetuados com a solução proposta. Este capítulo inclui medições de desempenho, casos detetados, e análise estatística dos dados classificados.

A apresentação dos resultados foi organizada em duas partes distintas com o objetivo de garantir uma análise mais clara, robusta e comparável. Esta divisão deve-se, principalmente, às grandes diferenças observadas no volume de validações entre dias de menor movimento (fins de semana, feriados) e dias de maior fluxo (dias úteis).

Agrupar todos os resultados num único conjunto teria dificultado a identificação de padrões específicos e introduzido vieses na análise estatística, especialmente no que diz respeito aos tempos de execução, proporção de classificações e desempenho do sistema sob diferentes cargas.

Ao separar os dados, tornou-se possível observar, de forma isolada, o comportamento do sistema em contextos menos exigentes (Parte 1) e em cenários mais complexos e intensivos (Parte 2), permitindo avaliar tanto a eficácia das regras quanto a escalabilidade do sistema.

Esta abordagem também proporciona *insights* mais precisos para futuras adaptações do sistema, nomeadamente na alocação de recursos computacionais e na otimização dos processos de limpeza, manipulação e classificação de dados.

5.1. Parte 1

Esta primeira parte dos testes centra-se na análise de dias com menor volume de validações, como fins de semana, feriados e alguns dias úteis com menor fluxo. O objetivo é avaliar o comportamento do sistema de classificação em cenários menos exigentes em termos de validações, mas ainda representativos de situações reais.

A Tabela 3 apresenta os resultados obtidos para este conjunto de dias, incluindo informações detalhadas sobre o número total de validações, os tempos de execução das etapas de limpeza, manipulação e classificação, bem como a distribuição das classificações atribuídas (fraudes, suspeitas, erros do sistema, inválidos e legítimos).

Esta análise permite identificar variações de desempenho do sistema em função do volume de dados e detetar eventuais padrões ou anomalias. Além disso, ajuda a reconhecer dias com maior incidência de eventos suspeitos ou classificações inesperadas, contribuindo para a melhoria contínua das regras de deteção.

É importante observar que os tempos de execução podem variar consoante o contexto operacional da máquina utilizada, uma vez que o sistema de regras e o servidor MySQL foram executados no mesmo ambiente local, estando sujeitos a possíveis interferências de outros processos em paralelo.

Data	Dados	Cleaning	Manipulation	Classification	Dados em Falta	Erro do sistema 4	Fraude 10	Inválido	Legítimo	Suspeita de Fraude 9	Suspeita de Fraude 5	Fraude / Erro do Sistema 6.1
2022-12-31	28458	00:01:16	00:10:06	00:21:48	1135	12	59	342	26604	306	0	0
2023-11-26	23731	00:01:14	00:07:11	00:15:26	1647	3	72	543	21267	199	0	0
2023-03-05	21025	00:01:10	00:05:27	00:12:06	925	2	57	219	19641	181	0	0
2023-12-30	31849	00:01:30	00:12:10	00:28:14	2256	16	47	878	28409	243	0	0
2023-01-22	19808	00:01:12	00:05:01	00:10:52	1915	2	51	575	17050	215	0	0
2023-06-11	21639	00:01:15	00:05:43	00:12:55	845	4	53	262	20332	137	1	0
2023-03-19	20603	00:01:30	00:05:13	00:11:39	1370	5	37	508	18504	169	1	9
2023-12-25	179	00:01:28	00:00:01	00:00:01	16	0	0	0	163	0	0	0
2023-04-16	22318	00:01:11	00:06:05	00:13:32	1639	2	35	538	19875	227	1	0
2023-12-10	24885	00:01:30	00:07:34	00:16:48	1555	4	56	390	22676	204	0	0
2023-08-05	30960	00:01:23	00:11:52	00:26:23	1602	8	45	551	27739	237	0	0
2023-03-12	22662	00:01:26	00:06:37	00:14:02	829	5	51	192	21403	182	0	0
2023-08-19	30520	00:01:25	00:11:56	00:25:19	7817	7	40	3476	18228	246	0	0

2023-04-06	36467	00:01:29	00:17:06	00:36:06	2657	6	47	1154	32232	354	0	0
2023-12-24	22646	00:01:31	00:06:14	00:13:56	1760	3	68	552	20030	230	3	0
2023-12-17	24489	00:01:16	00:07:19	00:16:21	1869	8	51	636	21745	180	0	0
2023-06-04	24630	00:01:25	00:07:22	00:16:27	2298	4	45	860	21242	172	2	0
2023-05-07	24610	00:01:28	00:07:18	00:16:28	1576	6	56	456	22285	219	1	0
2023-02-26	21454	00:01:28	00:05:34	00:12:28	1096	6	61	311	19814	166	0	0

Tabela 3 - Resultados da parte 1 dos testes

Como mencionado anteriormente, a Tabela 3 representa um conjunto de resultados obtidos durante a primeira fase do teste, relacionados a erros, fraudes e outras classificações do processo de classificação. Para facilitar a compreensão, apresentamos uma descrição das colunas que a compõem, destacando algumas observações relevantes.

Descrição das colunas:

- Data: Indica as datas associadas aos registos, correspondendo aos anos de 2022 e 2023.
- Dados: Refere-se ao número de validações realizadas em cada dia e, consequentemente, ao número de registos analisados pelo sistema de deteção de fraudes.
- *Cleaning*: O tempo dedicado à limpeza dos dados.
- *Manipulation*: O tempo dedicado à manipulação dos dados.
- *Classification*: O tempo dedicado à classificação dos dados.
- Dados em falta: Quantidade de validações que apresentam dados ausentes ou incompletos.
- Erros do sistema 4: Número de validações classificadas como “Erro do sistema 4”.
- Fraude 10: Quantidade de validações classificadas como “Fraude 10”.
- Inválido: Número de validações em que não é possível determinar se são fraude ou legítimas.
- Legítimo: Número de registos classificados como legítimos.
- Suspeita de Fraude 9: Quantidade de validações classificadas como “Suspeita de Fraude 9”.
- Suspeita de Fraude 5: Quantidade de registos classificados como “Suspeita de Fraude 5”.
- Fraude / Erro do Sistema 6.1: Número de validações classificadas como “Fraude / Erro do Sistema 6.1”.

Observações relevantes:

- Data com menor número de registos:
    - 25/12/2023: Este dia apresenta um número reduzido de registos (apenas 179), com a maioria das validações classificadas como legítimas (163) e nenhum caso de fraude ou erro identificado. A quantidade de "Dados em falta" foi de apenas 16, sugerindo que este dia não apresentou ameaças.
- Data com maior número de registos:
    - 06/04/2023: Este dia registou o maior número de validações, totalizando 36.467. A maioria das delas foi classificada como legítima (32.232), com 47 ocorrências de "Fraude 10" e 6 registos classificados como "Erro do Sistema 4". Adicionalmente, foram identificados 354 casos de "Suspeita de Fraude 9", e o número de registos com "Dados em Falta" foi de 2.657.
- Alta frequência de classificações não legítimas:
    - 26/11/2023: Destaca-se como o dia com o maior número de classificações como "Fraude 10" (72 registos).
    - 19/08/2023: Este foi o dia com o maior número de dados em falta, o que resultou no maior volume de registos classificados como "Inválido".
    - 30/12/2023: Foi o dia com a maior quantidade de classificações como "Erro do Sistema 4".
- Alta frequência de classificações legítimas:
    - Muitos registos, especialmente os de 30/12/2023 (com 28.409 registos legítimos), mostram uma expressiva quantidade de validações válidas, indicando que a maior parte das validações não apresentou indícios de fraude.

Conclusão:

- Observa-se uma variabilidade entre os dias, com algumas datas apresentando grandes quantidades de dados em falta e validações classificadas como erros ou fraudes, enquanto outras demonstram números mais reduzidos nesses aspetos. No entanto, mesmo nos dias com maior número de erros ou fraudes, a grande maioria das validações foi classificada como legítima, o que é um indicativo positivo

5.2. Parte 2

Esta secção apresenta os resultados da segunda parte dos testes, centrados em dias com elevado volume de validações — geralmente dias úteis —, o que permite avaliar o desempenho do sistema em contextos de maior número de registos.

A Tabela 4 compila um conjunto abrangente de informações relevantes para esta análise: o volume de dados processados, os tempos associados às diferentes fases do processamento (limpeza, manipulação e classificação), e as diversas classificações atribuídas (fraudes, suspeitas, erros, inválidos e legítimos).

Este nível de detalhe permite identificar padrões de comportamento nos dados, avaliar a consistência da performance do sistema ao longo do tempo e detetar eventuais *bottlenecks* ou falhas nos processos automáticos.

Deve ainda considerar-se que os tempos de execução podem ser afetados por fatores externos, como a execução simultânea de outros processos no mesmo computador, dado que o sistema de regras e o servidor MySQL estão alojados na mesma máquina.

Data	Dados	Cleaning	Manipulation	Classification	Dados em Falta	Erro do sistema 4	Fraude 10	Inválido	Legítimo	Suspeita de Fraude 9	Suspeita de Fraude 5	Fraude 6	Fraude 1
2023-07-17	53998	00:01:17	00:35:18	01:18:07	3317	17	89	1194	47277	494	0	0	0
2023-01-05	58367	00:01:33	00:40:23	01:30:03	4954	13	112	1440	51469	379	0	0	0
2023-08-17	50674	00:01:21	00:31:21	01:08:52	3131	20	69	908	44741	440	0	0	0
2023-04-03	51016	00:01:27	00:31:03	01:10:12	5591	10	72	2065	42858	376	0	0	0
2023-02-09	59183	00:01:28	00:41:42	01:35:28	4403	8	81	1378	52961	352	0	0	0
2023-09-15	60980	00:01:32	00:44:11	01:40:33	4811	14	81	1645	53949	480	0	0	0
2023-06-29	53341	00:01:24	00:34:26	01:16:10	4494	7	85	1792	46567	377	0	0	0
2023-10-12	61937	00:01:22	00:45:32	01:42:58	4958	11	98	1659	54733	478	0	0	0
2023-07-21	55381	00:01:30	00:37:04	01:22:39	3705	15	106	1246	48163	511	0	0	0
2023-05-08	60749	00:01:28	00:44:12	01:38:12	6426	12	80	2396	51374	439	0	0	0
2023-11-14	62140	00:01:27	00:46:00	01:46:26	6298	17	73	2005	53297	450	0	0	0
2023-08-03	46623	00:01:30	00:26:54	00:58:37	2856	21	71	945	40991	406	0	0	0
2023-10-27	60809	00:01:20	00:44:05	01:39:50	4336	29	95	1362	54478	509	0	0	0
2023-09-04	50443	00:01:31	00:30:27	01:07:54	2829	19	88	855	46202	450	0	0	0
2023-07-07	55259	00:01:29	00:36:24	01:22:02	4004	14	63	1465	47746	545	0	1	0
2023-08-31	50317	00:01:23	00:30:12	01:07:51	3932	8	75	1413	43153	404	0	0	2

2023-11-02	61365	00:01:21	00:44:48	01:41:20	5343	14	82	1848	53417	661	0	0	0
2023-02-13	59480	00:01:25	00:42:00	01:36:30	4120	13	91	948	53891	416	1	0	0
2023-10-06	61055	00:01:29	00:44:19	01:40:01	4325	9	124	1164	54860	573	0	0	0
2023-05-23	59437	00:01:21	00:42:49	01:35:20	7345	14	80	3092	48431	450	0	0	0
2023-09-18	60661	00:01:21	00:44:15	01:39:34	4203	11	121	1385	54564	377	0	0	0
2023-06-22	54903	00:01:30	00:36:05	01:22:32	6179	11	94	2463	45694	441	1	0	0
2023-01-30	56763	00:01:27	00:38:47	01:28:25	4060	9	94	1448	50817	335	0	0	0
2023-07-14	55456	00:01:20	00:36:40	01:21:56	4233	7	83	1766	47184	507	0	0	0
2023-05-25	58628	00:01:28	00:41:27	01:32:42	7376	12	101	3134	47572	409	0	0	0
2023-11-09	62122	00:01:29	00:46:32	01:43:33	4612	10	115	1631	55295	459	0	0	0
2023-05-17	61029	00:01:26	00:44:46	01:42:42	6774	12	97	2765	50694	658	0	0	0
2023-02-20	51693	00:01:28	00:32:02	01:13:43	4425	10	79	1633	45127	419	0	0	0
2023-06-07	54560	00:01:19	00:38:00	01:19:54	4443	6	97	2077	47543	390	0	0	0
2023-09-28	60660	00:01:28	00:46:14	01:39:21	16939	14	61	6460	36786	400	0	0	0
2023-10-02	60655	00:01:21	00:44:23	01:40:35	5405	13	62	1925	52762	488	0	0	0
2023-07-11	55557	00:01:32	00:37:28	01:24:32	4332	10	79	1706	47324	532	1	0	2
2023-03-15	60029	00:01:21	00:44:07	01:38:11	5602	6	88	1975	51941	417	0	0	0

2023-11-08	64131	00:01:32	00:49:58	01:51:07	5362	9	97	1809	56444	410	0	0	0
2023-06-27	50844	00:01:16	00:30:42	01:11:06	5973	13	104	2381	41756	593	0	0	0
2023-01-10	58396	00:01:25	00:40:36	01:32:55	5167	8	73	1657	50915	576	0	0	0
2023-08-22	51039	00:01:36	00:30:51	01:10:38	3213	17	77	1101	44743	468	0	0	0
2023-12-05	59311	00:01:32	00:41:49	01:34:19	5341	13	110	2005	51432	410	0	0	0
2023-03-01	58616	00:01:33	00:40:53	01:33:04	4982	14	97	1551	51408	563	0	0	0
2023-04-13	51220	00:01:31	00:31:09	01:10:52	4544	7	80	1693	44539	318	0	0	0
2023-10-30	61178	00:01:34	00:44:25	01:41:41	5204	23	90	1859	53543	459	0	0	0
2023-04-04	49985	00:01:30	00:29:52	01:08:15	5718	12	75	2097	41674	377	0	0	0
2023-08-07	52150	00:01:33	00:32:30	01:12:16	3481	14	108	1169	45601	392	0	0	0
2023-09-04	50443	00:01:14	00:30:36	01:08:07	2829	19	88	855	46202	450	0	0	0
2023-01-17	57870	00:01:29	00:39:49	01:29:11	5393	11	85	1641	50317	423	0	0	0

Tabela 4 - Resultados da parte 2 dos testes

Como mencionado anteriormente, a Tabela 4 representa um conjunto de resultados obtidos durante a primeira fase do teste, relacionados a erros, fraudes e outras classificações do processo de classificação. Para facilitar a compreensão, apresentamos uma descrição das colunas que a compõem, destacando algumas observações relevantes.

Descrição das colunas:

- Data: Indica as datas associadas aos registos, correspondendo aos anos de 2022 e 2023.
- Dados: Refere-se ao número de validações realizadas em cada dia e, consequentemente, ao número de registos analisados pelo sistema de deteção de fraudes.
- *Cleaning*, *Manipulation*, *Classification*: O tempo gasto em cada uma dessas fases de processamento.
- Dados em falta: Quantidade de validações que apresentam dados ausentes ou incompletos.
- Erros do sistema 4: Número de validações classificadas como "Erro do sistema 4".
- Fraude 10: Quantidade de validações classificadas como "Fraude 10".
- Inválido: Número de validações em que não é possível determinar se são fraude ou legítimas.
- Legítimo: Número de registos classificados como legítimos.
- Suspeita de Fraude 9: Quantidade de validações classificadas como "Suspeita de Fraude 9".
- Suspeita de Fraude 5: Quantidade de registos classificados como "Suspeita de Fraude 5".
- Fraude 6: Número de registos classificados como "Fraude 6".
- Fraude 1: Número de validações classificadas como "Fraude 1".

Observações relevantes:

- Volume e Tempo: O número de registos processados por dia varia significativamente, com picos até 64.131 validações (em 08/11/2023) e quedas para 46.623 validações (em 03/08/2023).
- Fraudes identificadas: A maioria dos dias não apresentam casos de "Fraude 6" e de "Fraude 1", mas há algumas exceções, como no dia 07/07/2023, 31/08/2023 e 11/07/2023, com algumas fraudes confirmadas.
- Dados em falta: Há dias com altos volumes de validações contendo dados ausentes, como no dia 28/09/2023, quando foram registadas 16.939 nessas condições. Esta situação irá impactar diretamente o número de registos classificados como inválidos.
- Erro do sistema: A classificação "Erro do sistema 4" ocorre de forma esporádica, com a maioria das validações apresentando contagens baixas, geralmente abaixo de 20 ocorrências.
- Suspeitas de fraude: A classificação "Suspeita de Fraude 9" é observada com maior frequência em comparação à "Suspeita de Fraude 5", com um número máximo de 661 ocorrências e um mínimo de 335 validações. Por outro lado, a classificação "Suspeita de Fraude 5" ocorre de forma rara, com uma quantidade mínima de uma ocorrência.

Destaques de casos:

- 06/10/2023: Neste dia, foi registado o maior número de casos classificados como "Fraude 10", totalizando 124 validações.

- 28/09/2023: Este dia apresentou o menor número de casos classificados como "Fraude 10", com 61 validações.
- 23/05/2023: Este foi o dia com o maior volume de registos para os quais não foi possível atribuir uma classificação, totalizando 3.092 validações.

Conclusão:

- Assim como observado na Tabela 3, persiste a variabilidade entre os dias, com algumas datas registando grandes volumes de dados ausentes e validações classificadas como erros ou fraudes, enquanto outras apresentam números mais baixos nesses aspetos. Contudo, mesmo nos dias com maior incidência de erros ou fraudes, a maioria das validações foi classificada como legítima, o que representa um indicativo positivo.

Para facilitar a leitura, as tabelas com os dados agregados e as respetivas observações são mantidas neste capítulo. No entanto, os gráficos detalhados correspondentes a esses resultados encontram-se no Anexo B – Representação Gráfica dos Resultados, permitindo assim uma leitura mais fluida sem perda de informação visual.

5.3. Análise estatística

Para realizar uma análise estatística dos resultados previamente apresentados, compilamos os dados das tabelas anteriores e efetuamos cálculos para identificar quais os dias e meses registaram o maior número de casos legítimos e não legítimos. É importante salientar que um dia com um maior número de uma determinada classificação não implica necessariamente que tenha sido o dia com maior incidência de fraude. Para obter uma avaliação precisa, dividimos o número de casos de cada classificação pelo tamanho da amostra correspondente, a fim de calcular o volume proporcional de casos de fraude ou não fraude em relação ao total de casos analisados.

Além disso, pretendemos identificar variações na frequência de casos legítimos e não legítimos ao longo do tempo. Esta análise permite observar não apenas picos isolados, mas também possíveis tendências sazonais ou períodos de maior vulnerabilidade, fornecendo uma visão mais completa sobre a distribuição temporal dos casos.

A tabela 5 ilustra exatamente o que foi descrito anteriormente, apresentando as variáveis reunidas e a relação entre o número de ocorrências de cada classificação e o tamanho da amostra correspondente. Cada coluna destaca em vermelho o valor máximo observado, permitindo uma visualização clara dos picos.

Notamos que, em muitos casos, o maior número absoluto de ocorrências numa classificação não corresponde ao maior valor proporcional, ou seja, ao valor resultante da divisão entre o número de ocorrências e o tamanho da amostra. Isso indica que mesmo dias com um alto volume de casos podem apresentar uma menor incidência proporcional, dependendo do tamanho da amostra. Portanto, esta análise proporcional revela-se essencial para uma avaliação precisa da prevalência de cada tipo de classificação.

Além disso, devido ao grande número de colunas, alguns títulos foram abreviados e o tamanho da fonte foi reduzida para otimizar o espaço disponível. Essas adaptações visaram manter a legibilidade e facilitar a interpretação dos dados, sem comprometer o conteúdo informativo de cada coluna.

Já a tabela 6, apresenta uma visão consolidada das ocorrências mensais, com as mesmas colunas e as relações proporcionais da tabela diária. Essa visão mensal permite identificar tendências ao longo do ano e avaliar períodos de maior incidência em cada classificação. Tal como na tabela diária, os títulos foram abreviados e o tamanho da fonte ajustado para uma apresentação mais organizada, garantindo que a informação se mantenha acessível e clara.

Data	Dados	Clean.	Manipu.	Classif.	Dados em Falta	Dados em Falta / Dados	Erro do sistema 4	Erro do sistema 4 / Dados	Fraude 10	Fraude 10 / Dados	Inválido	Inválido / Dados	Legítimo	Legítimo / Dados	Suspeita de Fraude 9	Suspeita de Fraude 9 / Dados	Suspeita de Fraude 5	Suspeita de Fraude 5 / Dados	Fraude 6	Fraude 6 / Dados	Fraude / Erro do Sistema 6.1	Fraude / Erro do Sistema 6.1 / Dados	Fraude 1	Fraude 1 / Dados
31/12/2022	28458	00:01:16	00:10:06	00:21:48	1135	0,039883337	12	0,000421674	59	0,002073231	26604	0,01201771	26604	0,93485136	306	0,010752688	0	0	0	0	0	0	0	0
05/01/2023	58367	00:01:33	00:40:23	01:30:03	4954	0,084876728	13	0,000222729	112	0,001918893	1440	0,024671475	51469	0,88181678	379	0,006493395	0	0	0	0	0	0	0	0
10/01/2023	58396	00:01:25	00:40:36	01:32:55	5167	0,088482088	8	0,000136996	73	0,001250086	1657	0,028375231	50915	0,87189191	576	0,009863689	0	0	0	0	0	0	0	0
17/01/2023	57870	00:01:29	00:39:49	01:29:11	5393	0,093191636	11	0,000190081	85	0,001468809	1641	0,028356661	50317	0,869483325	423	0,007309487	0	0	0	0	0	0	0	0
22/01/2023	19808	00:01:12	00:05:01	00:10:52	1915	0,09667811	2	0,000100969	51	0,002574717	17050	0,029028675	17050	0,860763328	215	0,0108542	0	0	0	0	0	0	0	0
30/01/2023	56763	00:01:27	00:38:47	01:28:25	4060	0,071525466	9	0,000158554	94	0,001656008	1448	0,025509575	50817	0,895248666	335	0,005901732	0	0	0	0	0	0	0	0
09/02/2023	59183	00:01:28	00:41:42	01:35:28	4403	0,074396364	8	0,000135174	81	0,001368636	1378	0,023283713	52961	0,894868459	352	0,005947654	0	0	0	0	0	0	0	0
13/02/2023	59480	00:01:25	00:42:00	01:36:30	4120	0,06926698	13	0,000218561	91	0,001529926	948	0,01593813	53891	0,906035642	416	0,006993948	1	1,68124E-05	0	0	0	0	0	0
20/02/2023	51693	00:01:28	00:32:02	01:13:43	4425	0,085601532	10	0,00019345	79	0,001528253	1633	0,031590351	45127	0,872980868	419	0,008105546	0	0	0	0	0	0	0	0
26/02/2023	21454	00:01:28	00:05:34	00:12:28	1096	0,051086045	6	0,000279668	61	0,002843293	19814	0,014496131	19814	0,923557379	166	0,007737485	0	0	0	0	0	0	0	0
01/03/2023	58616	00:01:33	00:40:53	01:33:04	4982	0,084993858	14	0,000238843	97	0,001654838	1551	0,026460352	51408	0,877030162	563	0,009604886	0	0	0	0	0	0	0	0
05/03/2023	21025	00:01:10	00:05:27	00:12:06	925	0,043995244	2	9,51249E-05	57	0,002711058	19641	0,010416171	19641	0,934173603	181	0,008608799	0	0	0	0	0	0	0	0
12/03/2023	22662	00:01:26	00:06:37	00:14:02	829	0,036581061	5	0,000220634	51	0,002250463	21403	0,008472333	21403	0,944444444	182	0,008031065	0	0	0	0	0	0	0	0
15/03/2023	60029	00:01:21	00:44:07	01:38:11	5602	0,093321561	6	9,99517E-05	88	0,001465958	1975	0,032900765	51941	0,865265122	417	0,006946642	0	0	0	0	0	0	0	0
19/03/2023	20603	00:01:30	00:05:13	00:11:39	1370	0,066495171	5	0,000242683	37	0,001795855	18504	0,024656603	18504	0,898121633	169	0,008202689	1	4,85366E-05	0	0	9	0,00043683	0	0
03/04/2023	51016	00:01:27	00:31:03	01:10:12	5591	0,109593069	10	0,000196017	72	0,001411322	2065	0,040477497	42858	0,840089384	376	0,007370237	0	0	0	0	0	0	0	0

04/04/2023	49985	00:01:30	00:29:52	01:08:15	5718	0,114394318	12	0,000240072	75	0,00150045	2097	0,041952586	41674	0,833730119	377	0,007542263	0	0	0	0	0	0	0	0
06/04/2023	36467	00:01:29	00:17:06	00:36:06	2657	0,072860394	6	0,000164532	47	0,001288836	32232	0,031645049	32232	0,883867606	354	0,009707407	0	0	0	0	0	0	0	0
13/04/2023	51220	00:01:31	00:31:09	01:10:52	4544	0,088715346	7	0,000136665	80	0,00156189	1693	0,033053495	44539	0,869562671	318	0,006208512	0	0	0	0	0	0	0	0
16/04/2023	22318	00:01:11	00:06:05	00:13:32	1639	0,07343848	2	8,96138E-05	35	0,001568241	19875	0,024106103	19875	0,890536786	227	0,010171162	1	4,48069E-05	0	0	0	0	0	0
07/05/2023	24610	00:01:28	00:07:18	00:16:28	1576	0,064039009	6	0,000243803	56	0,002275498	22285	0,018529053	22285	0,905526209	219	0,008898822	1	4,06339E-05	0	0	0	0	0	0
08/05/2023	60749	00:01:28	00:44:12	01:38:12	6426	0,105779519	12	0,000197534	80	0,001316894	2396	0,039440978	51374	0,845676472	439	0,007226456	0	0	0	0	0	0	0	0
17/05/2023	61029	00:01:26	00:44:46	01:42:42	6774	0,110996412	12	0,000196628	97	0,001589408	2765	0,04530633	50694	0,830654279	658	0,010781759	0	0	0	0	0	0	0	0
23/05/2023	59437	00:01:21	00:42:49	01:35:20	7345	0,123576224	14	0,000235544	80	0,001345963	3092	0,052021468	48431	0,814829147	450	0,007571042	0	0	0	0	0	0	0	0
25/05/2023	58628	00:01:28	00:41:27	01:32:42	7376	0,125810193	12	0,00020468	101	0,001722726	3134	0,053455687	47572	0,811421164	409	0,006976189	0	0	0	0	0	0	0	0
04/06/2023	24630	00:01:25	00:07:22	00:16:27	2298	0,093300853	4	0,000162404	45	0,00182704	21242	0,034916768	21242	0,862444174	172	0,006983354	2	8,12018E-05	0	0	0	0	0	0
07/06/2023	54560	00:01:19	00:38:00	01:19:54	4443	0,081433284	6	0,000109971	97	0,001777859	2077	0,038068182	47543	0,871389296	390	0,007148094	0	0	0	0	0	0	0	0
11/06/2023	21639	00:01:15	00:05:43	00:12:55	845	0,039049864	4	0,000184851	53	0,002449281	20332	0,012107768	20332	0,939599797	137	0,006331161	1	4,62129E-05	0	0	0	0	0	0
22/06/2023	54903	00:01:30	00:36:05	01:22:32	6179	0,112543941	11	0,000200353	94	0,00171211	2463	0,044860937	45694	0,832267818	441	0,008032348	1	1,82139E-05	0	0	0	0	0	0
27/06/2023	50844	00:01:16	00:30:42	01:11:06	5973	0,117476988	13	0,000255684	104	0,002045472	2381	0,046829518	41756	0,821257179	593	0,011663126	0	0	0	0	0	0	0	0
29/06/2023	53341	00:01:24	00:34:26	01:16:10	4494	0,084250389	7	0,000131231	85	0,001593521	1792	0,033595171	46567	0,873005755	377	0,007067734	0	0	0	0	0	0	0	0
07/07/2023	55259	00:01:29	00:36:24	01:22:02	4004	0,072458785	14	0,000253352	63	0,001140086	1465	0,026511518	47746	0,864040247	545	0,009862647	0	0	1	1,80966E-05	0	0	0	0
11/07/2023	55557	00:01:32	00:37:28	01:24:32	4332	0,077973973	10	0,000179995	79	0,001421963	1706	0,030707202	47324	0,851809853	532	0,009575751	1	1,79995E-05	0	0	0	0	2	3,59991E-05
14/07/2023	55456	00:01:20	00:36:40	01:21:56	4233	0,076330785	7	0,000126226	83	0,001496682	1766	0,031845066	47184	0,850836699	507	0,009142383	0	0	0	0	0	0	0	0
17/07/2023	53998	00:01:17	00:35:18	01:18:07	3317	0,061428201	17	0,000314826	89	0,001648209	1194	0,02211193	47277	0,875532427	494	0,009148487	0	0	0	0	0	0	0	0

21/07/2023	55381	00:01:30	00:37:04	01:22:39	3705	0,0669002	15	0,000270851	106	0,001914014	1246	0,022498691	48163	0,869666492	511	0,009226991	0	0	0	0	0	0	0	0
03/08/2023	46623	00:01:30	00:26:54	00:58:37	2856	0,061257319	21	0,000450421	71	0,001522854	945	0,020268966	40991	0,879201253	406	0,008708148	0	0	0	0	0	0	0	0
05/08/2023	30960	00:01:23	00:11:52	00:26:23	1602	0,051744186	8	0,000258398	45	0,001453488	27739	0,017797158	27739	0,895962532	237	0,007655039	0	0	0	0	0	0	0	0
07/08/2023	52150	00:01:33	00:32:30	01:12:16	3481	0,06674976	14	0,000268456	108	0,002070949	1169	0,022416107	45601	0,874419942	392	0,007516779	0	0	0	0	0	0	0	0
17/08/2023	50674	00:01:21	00:31:21	01:08:52	3131	0,06178711	20	0,00039468	69	0,001361645	908	0,017918459	44741	0,882918262	440	0,008682954	0	0	0	0	0	0	0	0
19/08/2023	30520	00:01:25	00:11:56	00:25:19	7817	0,25612713	7	0,000229358	40	0,001310616	18228	0,113892529	18228	0,597247706	246	0,008060288	0	0	0	0	0	0	0	0
22/08/2023	51039	00:01:36	00:30:51	01:10:38	3213	0,06295186	17	0,000333079	77	0,00150865	1101	0,021571739	44743	0,876643351	468	0,009169459	0	0	0	0	0	0	0	0
31/08/2023	50317	00:01:23	00:30:12	01:07:51	3932	0,078144563	8	0,000158992	75	0,00149055	1413	0,02808196	43153	0,857622672	404	0,008029096	0	0	0	0	0	0	2	3,9748E-05
04/09/2023	50443	00:01:31	00:30:27	01:07:54	2829	0,056083104	19	0,000376663	88	0,001744543	855	0,016949825	46202	0,915924905	450	0,00892096	0	0	0	0	0	0	0	0
04/09/2023	50443	00:01:14	00:30:36	01:08:07	2829	0,056083104	19	0,000376663	88	0,001744543	855	0,016949825	46202	0,915924905	450	0,00892096	0	0	0	0	0	0	0	0
15/09/2023	60980	00:01:32	00:44:11	01:40:33	4811	0,07889472	14	0,000229583	81	0,001328304	1645	0,026976058	53949	0,884699902	480	0,007871433	0	0	0	0	0	0	0	0
18/09/2023	60661	00:01:21	00:44:15	01:39:34	4203	0,069286692	11	0,000181336	121	0,001994692	1385	0,022831803	54564	0,899490612	377	0,006214866	0	0	0	0	0	0	0	0
28/09/2023	60660	00:01:28	00:46:14	01:39:21	16939	0,279244972	14	0,000230795	61	0,001005605	6460	0,106495219	36786	0,606429278	400	0,006594131	0	0	0	0	0	0	0	0
02/10/2023	60655	00:01:21	00:44:23	01:40:35	5405	0,089110543	13	0,000214327	62	0,001022175	1925	0,031736872	52762	0,86987058	488	0,008045503	0	0	0	0	0	0	0	0
06/10/2023	61055	00:01:29	00:44:19	01:40:01	4325	0,070837769	9	0,000147408	124	0,002030956	1164	0,019064778	54860	0,898534109	573	0,009384981	0	0	0	0	0	0	0	0
12/10/2023	61937	00:01:22	00:45:32	01:42:58	4958	0,080049082	11	0,0001776	98	0,001582253	1659	0,026785282	54733	0,883688264	478	0,007717519	0	0	0	0	0	0	0	0
27/10/2023	60809	00:01:20	00:44:05	01:39:50	4336	0,071305234	29	0,000476903	95	0,001562269	1362	0,022398	54478	0,895887122	509	0,008370471	0	0	0	0	0	0	0	0
30/10/2023	61178	00:01:34	00:44:25	01:41:41	5204	0,085063258	23	0,000375952	90	0,001471117	1859	0,03038674	53543	0,875200235	459	0,007502697	0	0	0	0	0	0	0	0
02/11/2023	61365	00:01:21	00:44:48	01:41:20	5343	0,087069176	14	0,000228143	82	0,001336267	1848	0,030114886	53417	0,870479915	661	0,010771612	0	0	0	0	0	0	0	0

08/11/2023	64131	00:01:32	00:49:58	01:51:07	5362	0,083610111	9	0,000140338	97	0,001512529	1809	0,028207887	56444	0,880135972	410	0,006393164	0	0	0	0	0	0	0	0
09/11/2023	62122	00:01:29	00:46:32	01:43:33	4612	0,07424101	10	0,000160974	115	0,001851196	1631	0,026254789	55295	0,890103345	459	0,007388687	0	0	0	0	0	0	0	0
14/11/2023	62140	00:01:27	00:46:00	01:46:26	6298	0,101351786	17	0,000273576	73	0,001174767	2005	0,032265851	53297	0,857692308	450	0,007241712	0	0	0	0	0	0	0	0
26/11/2023	23731	00:01:14	00:07:11	00:15:26	1647	0,069402891	3	0,000126417	72	0,003034006	21267	0,022881463	21267	0,896169567	199	0,008385656	0	0	0	0	0	0	0	0
05/12/2023	59311	00:01:32	00:41:49	01:34:19	5341	0,090050749	13	0,000219184	110	0,001854631	2005	0,033804859	51432	0,867157863	410	0,006912714	0	0	0	0	0	0	0	0
10/12/2023	24885	00:01:30	00:07:34	00:16:48	1555	0,062487442	4	0,000160739	56	0,002250352	22676	0,015672092	22676	0,911231666	204	0,008197709	0	0	0	0	0	0	0	0
17/12/2023	24489	00:01:16	00:07:19	00:16:21	1869	0,07631998	8	0,000326677	51	0,002082568	21745	0,025970844	21745	0,887949692	180	0,007350239	0	0	0	0	0	0	0	0
24/12/2023	22646	00:01:31	00:06:14	00:13:56	1760	0,077717919	3	0,000132474	68	0,003002738	20030	0,024375166	20030	0,884482911	230	0,010156319	3	0,000132474	0	0	0	0	0	0
25/12/2023	179	00:01:28	00:00:01	00:00:01	16	0,089385475	0	0	0	0	163	0	163	0,910614525	0	0	0	0	0	0	0	0	0	0
30/12/2023	31849	00:01:30	00:12:10	00:28:14	2256	0,070834249	16	0,000502371	47	0,001475714	28409	0,027567585	28409	0,891990329	243	0,007629753	0	0	0	0	0	0	0	0

Tabela 5 - Análise das classificações diárias com proporções relativas ao tamanho da amostra

A Tabela 5 apresenta registos de dados processados em diferentes datas, detalhando etapas de tratamento e de classificação, como também erros, suspeitas e fraudes detetadas. As colunas incluem informações sobre as quantidades de dados, tempos de processamento e proporções.

Estrutura e colunas:

- Data: Indica as datas associadas aos registos, correspondendo aos anos de 2022 e 2023.
- Dados: Total de registos processados em cada dia.
- *Clean*., *Manipu*., *Classif*.: Tempos gastos nas etapas de limpeza, manipulação e classificação dos dados.
- Dados em falta: Número absoluto de registos com dados ausentes.
- Dados em falta / Dados: Proporção de registos com dados ausentes em relação ao total processado.
- Erro do Sistema 4 e Proporção: Quantidade de "Erros do Sistema 4" detetados e a sua proporção em relação ao total de dados nesse dia.
- Fraudes e Suspeitas de Fraude: Diferentes classificações, mas cada uma com seu valor absoluto e proporcional representado na tabela.

Análise Quantitativa:

- Volume de dados processados:
    - O número de registos processados varia significativamente, com valores de 172 até 64131 numa única data.
- Tempo de processamento:
    - Limpeza: Varia entre 00:01:12 a 00:01:36.
    - Manipulação: Varia entre 00:00:01 a 00:49:58.
    - Classificação: varia entre 00:00:01 a 01:51:07.
- Dados em falta:
    - A quantidade da classificação "Dados em Falta" variou entre 16 e 16939.
    - A maior proporção foi de 0,279244972 em 28/09/2023.
- Erros do Sistema 4:
    - A quantidade da classificação "Erro do Sistema 4" foi relativamente baixa, variando entre 0 e 29, com uma proporção máxima de 0,000502371.
- Fraudes e Suspeitas:
    - As fraudes e suspeitas de fraude foram registadas em diversas classificações, mas sempre com valores geralmente baixos. Em alguns dias, as suspeitas ultrapassaram 500 registos, como em 02/11/2023 (661 casos). Já as fraudes, em alguns dias, chegaram a pouco mais de 100 registos, como em 06/10/2023 (124 casos).

Destaques e casos específicos:

- 06/04/2023:

    - Número reduzido de classificações de "Fraude 10" (47).
    - 32.232 registos sem classificação concreta (máximo da tabela).
    - Considerável número de suspeitas de fraude (354).
- 08/11/2023:
    - Volume elevado de registos (64.131, máximo da tabela).
    - Alto tempo de manipulação dos dados (00:49:58).
    - Alto tempo de classificação (01:51:07).
    - Elevado número de classificações legítimas (56.444).
- 12/03/2023:
    - Número reduzido de erros do sistema (5).
    - Maior proporção de classificações legítimas (0,944444444, máximo da tabela).
- 28/09/2023:
    - Volume elevado de dados a serem classificados (60.660).
    - Número reduzido de "Erros do sistema 4" e "Fraudes 10" (14 e 61, respetivamente).
    - Maior proporção de dados em falta, tanto em absoluto quanto em proporção (16.939 e 0,279244972, ambos máximos da tabela).

Mês	Dados	Clean.	Manipu.	Classif.	Dados em Falta	Dados em Falta / Dados	Erro do sistema 4	Erro do sistema 4 / Dados	Fraude 10	Fraude 10 / Dados	Inválido	Inválido / Dados	Legítimo	Legítimo / Dados	Suspeita de Fraude 9	Suspeita de Fraude 9 / Dados	Suspeita de Fraude 5	Suspeita de Fraude 5 / Dados	Fraude 6	Fraude 6 / Dados	Fraude / Erro do Sistema 6.1	Fraude / Erro do Sistema 6.1 / Dados	Fraude 1	Fraude 1 / Dados
Jan.	251204	00:07:06	02:44:36	06:11:26	21489	0,0855 4402	43	0,00017 1176	415	0,00165 2044	23236	0,09249 8527	220568	0,87804 3343	1928	0,0076750 37	0	0	0	0	0	0	0	0
Fev.	191810	00:05:49	02:01:18	04:38:09	14044	0,0732 18289	37	0,00019 2899	312	0,00162 661	23773	0,12394 0358	171793	0,89564 152	1353	0,0070538 55	1	5,21349E -06	0	0	0	0	0	0
Mar.	182935	00:07:00	01:42:17	03:49:02	13708	0,0749 3372	32	0,00017 4926	330	0,00180 3919	63074	0,34478 9133	162897	0,89046 3826	1512	0,0082652 31	1	5,46642E -06	0	0	9	4,91978 E-05	0	0
Abr.	211006	00:07:08	01:55:15	04:18:57	20149	0,0954 90176	37	0,00017 535	309	0,00146 4413	57962	0,27469 3611	181178	0,85863 9091	1652	0,0078291 61	1	4,7392E- 06	0	0	0	0	0	0
Mai.	264453	00:07:11	03:00:32	06:45:24	29497	0,1115 39669	56	0,00021 1758	414	0,00156 5496	33672	0,12732 6973	220356	0,83325 2033	2175	0,0082245 24	1	3,78139E -06	0	0	0	0	0	0
Jun.	259917	00:08:09	02:32:18	05:39:04	24232	0,0932 29762	45	0,00017 3132	478	0,00183 9049	50287	0,19347 3301	223134	0,85848 1746	2110	0,0081179 76	4	1,53895E -05	0	0	0	0	0	0
Jul.	275651	00:07:08	03:02:54	06:49:16	19591	0,0710 71754	63	0,00022 855	420	0,00152 3666	7377	0,02676 2101	237694	0,86230 0518	2589	0,0093923 11	1	3,62778E -06	1	3,62778 E-06	0	0	2	7,25555 E-06
Ago.	312283	00:10:11	02:55:36	06:29:56	26032	0,0833 60285	95	0,00030 4211	485	0,00155 3078	51503	0,16492 4123	265196	0,84921 6896	2593	0,0083033 66	0	0	0	0	0	0	2	6,40445 E-06
Set.	283187	00:07:06	03:15:43	07:15:29	31611	0,1116 25887	77	0,00027 1905	439	0,00155 0212	11200	0,03954 9838	237703	0,83938 5283	2157	0,0076168 75	0	0	0	0	0	0	0	0
Out.	305634	00:07:06	03:42:44	08:25:05	24228	0,0792 71285	85	0,00027 811	469	0,00153 4515	7969	0,02607 367	270376	0,88463 9798	2507	0,0082026 21	0	0	0	0	0	0	0	0
Nov.	273489	00:07:03	03:14:29	07:17:52	23262	0,0850 56437	53	0,00019 3792	439	0,00160 5183	28560	0,10442 8332	239720	0,87652 5198	2179	0,0079674 14	0	0	0	0	0	0	0	0

Dez.	191817	00:10:03	01:25:13	03:11:27	13932	0,0726 31727	56	0,00029 1945	391	0,00203 8401	121632	0,63410 4381	171059	0,89178 2272	1573	0,0082005 24	3	1,56399E -05	0	0	0	0	0	0

Tabela 6 - Análise das classificações mensais com proporções relativas ao tamanho da amostra

A Tabela 6 resume um histórico detalhado do processamento de dados, desde as etapas iniciais até à deteção de irregularidades. A tabela quantifica os dados, mede o tempo de cada etapa e apresenta as proporções de erros, suspeitas e fraudes identificados.

Estrutura e colunas:

- Data: Apresenta as datas específicas de cada registo, abrangendo os anos de 2022 e 2023.
- Dados: Contagem de registos processados
- *Clean*., *Manipu*., *Classif*.: Duração do processo de limpeza, manipulação e classificação dos dados.
- Dados em falta: Número absoluto de registos com dados ausentes.
- Dados em falta / Dados: Taxa de registos com dados em falta, tendo em conta o total processado.
- Erro do Sistema 4 e Proporção: Quantidade de vezes que o "Erro do Sistema 4" ocorreu e a sua proporção em relação ao total de dados processados
- Fraudes e Suspeitas de Fraude: Diferentes classificações, com os seus valores absolutos individuais e a proporção que representam do total.

Análise Quantitativa:

- Volume de dados processados:
    - O número de registos processados varia significativamente, com valores de 191810 até 312283 num único mês.
- Tempo de processamento:
    - Limpeza: Varia entre 00:05:49 a 00:10:11.
    - Manipulação: Varia entre 01:25:13 a 03:42:44.
    - Classificação: varia entre 03:11:27 a 08:25:05.
- Dados em falta:
    - A quantidade da classificação "Dados em Falta" variou entre 13708 e 31611.
    - A maior proporção foi de 0,111625887 em setembro.
- Erros do Sistema 4:
    - A quantidade da classificação "Erro do Sistema 4" foi relativamente baixa, variando entre 32 e 95, com uma proporção máxima de 0,000304211.
- Fraudes e Suspeitas:
    - As fraudes e suspeitas de fraude foram registadas em diversas classificações, mas sempre com valores geralmente baixos. Em alguns dias, as suspeitas ultrapassaram 2000 registos, como em agosto (2593 casos). Já as fraudes, em alguns dias, chegaram a pouco mais de 400 registos, como em agosto (485 casos).

Destaques e casos específicos:

- Agosto:

    - Número elevado de classificações como "Fraude 10", totalizando 485 ocorrências, o maior valor registado na tabela.
    - Total de 31.2283 registos processados, o valor máximo presente na tabela.
    - Tempo de limpeza elevado, atingindo 00:10:11, o maior tempo registado na tabela.
    - Número significativo de ocorrências de "Erros do Sistema 4", totalizando 95, o maior valor da tabela.
    - Maior proporção de classificações como "Erro do Sistema 4", correspondendo a 0,000304211, o valor mais alto na tabela.
    - Quantidade elevada de ocorrências de "Suspeitas de Fraude 9", com um total de 2.593, o maior valor registado na tabela.
- Dezembro:
    - Maior proporção de classificações como "Fraude 10", correspondendo a 0,002038401, o valor mais elevado da tabela.
    - Número significativo de classificações como "Inválido", totalizando 121.632 ocorrências, o maior valor registado na tabela.
    - Maior proporção de registos com dados em falta, atingindo 0,634104381, o valor mais alto da tabela.
- Fevereiro:
    - Mês com a menor quantidade de registos processados, totalizando 191.810 ocorrências.
    - Maior proporção de registos classificados como "Legítimos", correspondendo a 0,89564152, o valor mais elevado da tabela.
- Outubro:
    - Tempo elevado de manipulação dos dados, totalizando 03:42:44, o valor mais alto da tabela.
    - Tempo elevado de classificação dos dados, atingindo 08:25:05, o maior registado na tabela.
    - Número significativo de registos classificados como "Legítimo", totalizando 270.376 ocorrências, o valor máximo da tabela.

5.4. Automatização

Para facilitar o uso deste sistema, foi realizada uma modificação nas transformações, especificamente no *Data Cleaning* e *Data Manipulation*, de forma que as três transformações sejam executadas com um único clique. Por outras palavras, basta executar a transformação *Data Cleaning*, que ao concluir, automaticamente iniciará a transformação *Data Manipulation*, e no final desta, a transformação *Data Classification* será executada automaticamente. Para isso, foram introduzidos os steps “EXECUTE DATA MANIPULATION” e “EXECUTE DATA CLASSIFICATION”.

Anteriormente, durante a fase de testes, era necessário executar manualmente as três transformações de forma sequencial. Este processo manual permitia um maior controlo sobre a execução das transformações, pois elas só eram iniciadas quando o utilizador desejava. Consequentemente, era mais fácil de identificar onde estavam os erros quando ocorriam e também simplificava a obtenção de métricas, como o tempo de execução de cada transformação.

Contudo, com a automatização implementada, ainda é possível reverter para a execução manual. Basta desativar a conexão entre o *step* que bloqueia o fluxo e o que executa a transformação seguinte. As imagens 16, 17 e 18 ilustram de forma mais clara e visual a implementação da automatização nas transformações.

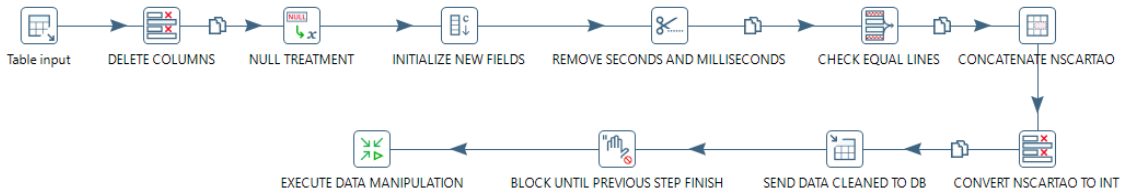


Imagem 16 - *Data Cleaning* com automatização

Imagem 17 - *Data Manipulation* com automatização

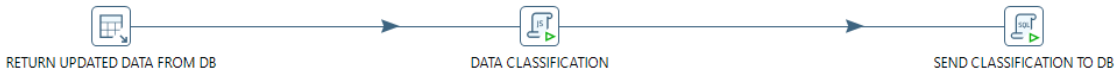


Imagem 18 - *Data Cleaning* com automatização

6. Conclusão

A presente dissertação teve como propósito estudar e propor uma solução prática para a deteção de fraudes em sistemas de bilhética eletrónica, com base no caso real da empresa Horários do Funchal. A investigação surgiu da constatação de que, apesar da digitalização dos transportes públicos, persistem fragilidades nos sistemas de controlo de validações, abrindo espaço à ocorrência de comportamentos ilícitos que resultam em perdas financeiras, quebra de confiança e desigualdade no acesso aos serviços.

Inicialmente, foi realizada uma revisão exaustiva da literatura, que permitiu compreender os diferentes tipos de fraude associados à bilhética eletrónica, desde a evasão de tarifas até à falsificação de bilhetes digitais. Esta revisão evidenciou não apenas a complexidade do problema, mas também as limitações das abordagens existentes, justificando a adoção de uma metodologia baseada em regras — mais transparente, replicável e ajustada à realidade do problema estudado.

No desenvolvimento do projeto, foi implementado um sistema de deteção de fraudes recorrendo a ferramentas de código aberto (Pentaho Data Integration e MySQL), com integração de um processo automatizado de ETL. Foram definidas treze regras de negócio específicas, concebidas para identificar padrões anómalos em dados históricos de validações. Cada uma destas regras foi testada individualmente e em conjunto, com dados reais e simulados, permitindo avaliar a robustez e a eficácia do sistema em diferentes cenários.

Os resultados obtidos confirmam a capacidade do sistema em classificar corretamente os registos como legítimos, inválidos ou fraudulentos. O processo de teste demonstrou uma deteção eficaz de comportamentos suspeitos, reduzindo falsos positivos e falsos negativos, e mostrando que a abordagem proposta é viável e útil na prática. A análise estatística dos resultados forneceu indicadores quantitativos relevantes, que poderão ser utilizados como suporte à tomada de decisão por parte das empresas de transporte.

Além disso, a solução desenvolvida apresenta diversas vantagens práticas: permite a automatização do processo de deteção, é escalável para outros contextos operacionais e pode ser facilmente atualizada com novas regras ou adaptada a diferentes realidades. O uso de tecnologias *open source* garante um custo reduzido de implementação e manutenção, o que a torna acessível mesmo para operadores de menor dimensão.

Apesar dos resultados encorajadores, este trabalho reconhece as suas limitações. O sistema baseado em regras, embora eficaz, depende da definição explícita de padrões suspeitos e pode não capturar formas de fraude mais sofisticadas ou emergentes. Assim, uma das principais direções futuras será a incorporação de técnicas de *machine learning*, que permitirão ao sistema aprender com novos dados e adaptar-se automaticamente a novas formas de fraude.

Em suma, esta dissertação contribui para a mitigação de um problema real e relevante nos transportes públicos, oferecendo uma solução prática, testada e replicável. Representa também um ponto de partida para investigações futuras mais ambiciosas, abrindo caminho para sistemas inteligentes de deteção de fraudes mais robustos, proativos e autossustentáveis.

6.1. Trabalho Futuro

Para dar continuidade e aprimorar este trabalho, propomos, em primeiro lugar, realizar mais testes em dias que ainda não foram analisados. Isto visa obter mais dados estatísticos mais precisos e verificar se, de facto, as tendências e os p-valor alteram-se ou não. Além disso, caso alguém conheça outros métodos mais avançados de análise de dados, recomendamos a sua exploração para proporcionar uma compreensão e análise mais profunda dos padrões observados e fortalecerão a robustez das conclusões apresentadas.

Em segundo lugar, considerando que o objetivo principal do projeto foi alcançado e o tempo limitado disponível para concluir este trabalho, torna-se inviável aprofundar no campo do *machine learning* neste momento. Contudo, sem dúvida, a incorporação de técnicas dessa matéria elevaria o sistema de deteção de fraude e o próprio trabalho a um nível superior, considerando que essa é uma área extremamente atual e poderosa. Um sistema de deteção de fraudes baseado em *machine learning* teria a capacidade de identificar fraudes mais complexas, algo que os sistemas baseados em regras não conseguem realizar ou com muita dificuldade. Por outras palavras, desenvolver um sistema baseado em *machine learning* utilizando como ponto de partida o trabalho realizado até agora, proporcionaria uma melhoria significativa na eficácia e precisão do sistema. Esse avanço não apenas aumentaria a capacidade de deteção de fraudes mais sutis e sofisticadas, mas também contribuiria para a robustez e adaptabilidade do sistema a novos padrões de comportamento fraudulento. Portanto, é altamente recomendado a exploração futura dessa matéria.

Em terceiro lugar, planeia-se realizar uma análise estatística mais abrangente e detalhada, incorporando não apenas um maior número de amostras, mas também diversificando os tipos de análises realizadas. Por exemplo, será possível identificar outros dados relevantes, como o tipo de bilhete ou passe associado ao maior número de casos de uso indevido. Com isto busca-se explorar outras visões analíticas que possam contribuir para uma compreensão mais profunda.

7. Referências

[1] Investopedia Staff. What Is Fraud? Definition, Types, and Consequences. Investopedia, 2023. Disponível em: https://www.investopedia.com/terms/f/fraud.asp. Acesso em: 20 set. 2023.

[2] Cydni, autor desconhecido. Fraud - Definition, Meaning, Types, and Examples. Legal Dictionary, 2023. Disponível em: https://legaldictionary.net/fraud/. Acesso em: 20 set. 2023.

[3] DOIG, A. Fraud. London: Taylor & Francis, 2006.

[4] FIROZABADI, B.; TAN, Y.-H.; LEE, R. Formal Definitions of Fraud, set. 2023. [Artigo].

[5] PIMENTA, C. Esboço de Quantificação da Fraude em Portugal. Work. Pap., 2009.

[6] MOURA, H.; SILVA, A. Auditoria de fraude: Instrumentos na prevenção de fraudes contra as empresas. Instituto Politécnico de Lisboa. Disponível em: https://repositorio.ipl.pt/bitstream/10400.21/1655/1/Auditoria%20de%20fraude.pdf. Acesso em: 20 set. 2023.

[7] Resumos Só Escola. Perjúrio: O que é, significado. Disponível em: https://resumos.soescola.com/glossario/perjurio-o-que-e-significado/. Acesso em: 20 set. 2023.

[8] SANTOS, F. O IMPACTO DA AUDITORIA INTERNA NA DETEÇÃO E PREVENÇÃO DE FRAUDE NAS EMPRESAS. 2022. Disponível em: https://recipp.ipp.pt/bitstream/10400.22/20947/1/Filipa%20Santos_MA_2022.pdf. Acesso em: 20 set. 2023.

[9] FÜRST, E. Free Riders and Ticket Fraud in Public Transport: A Delphi Analysis, 2012.

[10] MAYES, K. E.; MARKANTONAKIS, K.; HANCKE, G. Transport ticketing security and fraud controls. Information Security Technical Report, v. 14, n. 2, p. 87–95, maio 2009. DOI: 10.1016/j.istr.2009.06.003.

[11] BBC News Brasil. Como nasceu o primeiro sistema de transporte coletivo do mundo. 24 set. 2018. Disponível em: https://www.bbc.com/portuguese/geral-45587611. Acesso em: 7 out. 2023.

[12] MENDES LÜBECK, R.; WITTMANN, M. L.; FLORES BATTISTELLA, L. Electronic Ticketing System as a Process of Innovation. Journal of Technology Management & Innovation, v. 7, n. 1, p. 17–30, mar. 2012. DOI: 10.4067/S0718-27242012000100002.

[13] Luxiong Xu, Na Wang, Chenglian Liu. Security of electronic ticketing. IEEE Xplore, 2010. Disponível em: https://ieeexplore.ieee.org/abstract/document/5543314. Acesso em: 26 set. 2023.

[14] NordVPN Team. O que é o bloqueio RFID e como funciona. NordVPN Blog, 2023. Disponível em: https://nordvpn.com/pt/blog/tecnologia-rfid/. Acesso em: 20 set. 2023.

[15] NortonLifeLock Inc. 10 Ticketmaster scams and how to spot them. Norton Blog, 2023. Disponível em: https://us.norton.com/blog/online-scams/ticketmaster-scams. Acesso em: 3 out. 2023.

[16] RAMZIPOR, Evan. Online Ticketing Fraud: Challenges & Opportunities. Sift Blog, 2023. Disponível em: https://blog.sift.com/online-ticketing-fraud/. Acesso em: 3 out. 2023.

[17] HUTCHINGS, A. Leaving on a jet plane: the trade in fraudulently obtained airline tickets. Crime, Law and Social Change, v. 70, n. 4, p. 461–487, nov. 2018. DOI: 10.1007/s10611-018-9777-8.

[18] United States Department of Justice. West African computer hacker sentenced to Federal Prison. Northern District of Georgia, 2023. Disponível em: https://www.justice.gov/usao-ndga/pr/west-african-computer-hacker-sentenced-federal-prison. Acesso em: 27 set. 2023.

[19] DURGUT, M. PNR - Passenger Name Record - aviationfile. Disponível em: https://www.aviationfile.com/pnr-passenger-name-record/. Acesso em: 29 set. 2023.

[20] FARINELLI, B. Online Fraud Risk Profile for the Ticketing Industry. ClearSale Blog, 2023. Disponível em: https://blog.clear.sale/online-fraud-risk-profile-for-the-ticketting-industry. Acesso em: 7 out. 2023.

[21] KOU, Y.; LU, C.-T.; SIRWONGWATTANA, S.; HUANG, Y.-P. Survey of fraud detection techniques. Proceedings of IEEE International Conference on Networking, Sensing and Control, p. 749-754, vol. 2, mar. 2004. DOI: 10.1109/ICNSC.2004.1297040.

[22] Section.io Engineering Education Program. The Basics of Fraud Detection Analytics. [Online; página indisponível em 2025]. Disponível originalmente em: https://www.section.io/engineering-education/basics-of-fraud-detection-analytics/. Acesso em: 11 out. 2023.

[23] Wikipedia contributors. Unstructured data. Wikipedia, The Free Encyclopedia, 3 out. 2023. Disponível em: https://en.wikipedia.org/w/index.php?title=Unstructured_data&oldid=1178453747. Acesso em: 11 out. 2023.

[24] Fraud.net. Rules-Based Fraud Detection. Disponível em: https://fraud.net/d/rules-based-fraud-detection/. Acesso em: 21 out. 2023.

[25] ZHU, X. et al. Intelligent financial fraud detection practices in post-pandemic era. The Innovation, v. 2, n. 4, p. 100176, nov. 2021. DOI: 10.1016/j.xinn.2021.100176.

[26] OLIVEIRA, P. H. M. A. Detecção de fraudes em cartões: um classificador baseado em regras de associação e regressão logística. Universidade de São Paulo, 2015. DOI: 10.11606/D.45.2016.tde-01022016-204144.

[27] Plaid. Algorithmic and rules-based fraud models. Disponível em: https://fin.plaid.com/articles/algorithmic-and-rules-based-fraud-models/. Acesso em: 4 nov. 2023.

[28] SEON. Guide to Fraud Detection Rules & How to Choose a Solution. Disponível em: https://seon.io/resources/guides/guide-to-fraud-detection-rules/. Acesso em: 29 nov. 2023.

[29] Wikipédia. Pentaho. Última modificação 30 abr. 2020. Disponível em: https://pt.wikipedia.org/w/index.php?title=Pentaho&oldid=58156721. Acesso em: 26 nov. 2023.

[30] Talend. Talend Data Integration — Software to Connect, Access, and Transform Data. Disponível em: https://www.talend.com/products/integrate-data/. Acesso em: 26 nov. 2023.

[31] user2544374. Talend csv to relational db tables: foreign key setting. Stack Overflow, 2017. Disponível em: https://stackoverflow.com/q/45459944. Acesso em: 29 nov. 2023.

[32] Pentaho Community Wiki. Modified Java Script Value - Pentaho Data Integration. Disponível em: https://pentaho-public.atlassian.net/wiki/spaces/EAI/pages/371558212/Modified+Java+Script+Value. Acesso em: 7 nov. 2024.

8. Anexo A - Testes das Regras 3 a 13

Este anexo apresenta em detalhe os testes correspondentes às regras de deteção de fraude que não foram incluídas no corpo principal do capítulo 4.4, nomeadamente as regras terceira a quinta e sétima a décima terceira.

Estas regras foram testadas de forma individual, seguindo o mesmo procedimento descrito no início do capítulo 4.4:

- Manipulação controlada da base de dados (inserção ou alteração de registos);
- Execução do sistema para aplicação das regras de deteção;
- Verificação dos resultados através de *queries* de validação, com foco nas classificações atribuídas.

Cada secção deste anexo inclui o cenário de teste criado, as ações realizadas, o código utilizado, os resultados esperados e os resultados obtidos. Esta abordagem garante a rastreabilidade de cada regra e comprova a sua eficácia na identificação de comportamentos fraudulentos ou inválidos.

A inclusão destas regras neste anexo tem como objetivo não sobrecarregar a leitura principal, preservando ao mesmo tempo a integridade técnica e o rigor da validação de cada regra.

8.1. Terceira regra

Para avaliar a eficácia da terceira regra, foi desenvolvida uma *query* que copia 1000 linhas da tabela num intervalo definido, semelhantemente à primeira regra, onde é aplicada modificações específicas. Entre elas, destaca-se a aleatoriedade na escolha das descrições entre "Pré-comprado Viagens – Adulto", "Pré-comprado Grupo – Adulto", "Pré-comprado Grupo – Criança" e "Pré-comprado Viagens – Criança". Adicionalmente, a *query* ainda realiza alterações na coluna "Titulo" para que o valor corresponda às descrições selecionadas. Para desafiar a regra de classificação, os valores das colunas "TempoViagemRestanteMins" e "CounterValueBefore" são definidos como zero, o que, de acordo com a regra, deve resultar na classificação "Fraude 3". Além disso, foram efetuadas modificações nas colunas "Linha" e "Codigo" para garantir a unicidade dos valores, evitando duplicações.

O código 9 em MySQL realiza uma série de operações na tabela "dataoutput", inserindo novos registos, atualizando valores na coluna "Titulo" e ajustando as descrições na coluna "Descricao".

O primeiro bloco é responsável por inserir 1.000 novos registos na tabela "dataoutput" usando registos existentes como base:

- A consulta **SELECT** extrai registos existentes com "Linha" entre 1035005530 e 1035586691, limitando-se a 1.000 linhas.
- Para cada novo registo, as colunas "Linha" e "Codigo" recebem um valor exclusivo ao adicionar 1 ao valor máximo atual nessas colunas.
- Algumas colunas específicas ("TempoViagemRestanteMins" e "CounterValueBefore") são configuradas com valores fixos ("0").

O segundo bloco ajusta o valor da coluna “Titulo” com base no valor da coluna “Linha”:

- Usa o operador % (módulo) para definir valores de “Titulo”:
  - Se “Linha” é divisível por 3, “Titulo” é “4876”.
  - Se o resto da divisão é 1, “Titulo” é “4878”.
  - Se o resto é 2, “Titulo” é “4887”.
  - Para outros casos, “Titulo” seria “4885”.

Relativamente ao terceiro bloco, para cada valor específico de “Titulo”, “Descricao” recebe uma descrição correspondente:

- Titulo 4876: "Pré-comprado Viagens - Adulto".
- Titulo 4878: "Pré-comprado Grupo - Adulto".
- Titulo 4887: "Pré-comprado Grupo - Criança".
- Qualquer outro valor: "Pré-comprado Viagens - Criança"

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
  DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, 0 AS TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  0 AS CounterValueBefore, CounterValueAfter, Titulo,
  GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
LIMIT 1000;
```

```
update dataoutput set Titulo = case
   WHEN (Linha % 3) = 0 THEN 4876
   WHEN (Linha % 3) = 1 THEN 4878
   WHEN (Linha % 3) = 2 THEN 4887
   ELSE 4885
end;

update dataoutput set Descricao = case
         WHEN Titulo = 4876 then "Pré-comprado Viagens - Adulto"
   WHEN Titulo = 4878 then "Pré-comprado Grupo - Adulto"
   when Titulo = 4887 then "Pré-comprado Grupo - Criança"
   ELSE "Pré-comprado Viagens - Criança"
end;
```

Código 9 - Inserção de novos dados (3ª Regra)

A primeira *query*, no código 10, tem como objetivo selecionar registos onde "Titulo" seja um dos valores especificados (4876, 4878, 4885, 4887), e onde "TempoViagemRestanteMins" e "CounterValueBefore" são ambos iguais a "0". Por outras palavras, a condição **WHERE** filtra apenas registos com "Titulo" correspondente e onde tanto "TempoViagemRestanteMins" quanto "CounterValueBefore" possuem o valor 0.

A segunda *query* tem um objetivo similar à primeira consulta, mas agora retorna registos onde "TempoViagemRestanteMins" ou "CounterValueBefore" são diferentes de "0". Ou seja, a condição **WHERE** ainda filtra os mesmos valores de "Titulo", mas agora busca registos onde pelo menos uma dessas duas colunas ("TempoViagemRestanteMins" ou "CounterValueBefore") possui um valor diferente de 0.

```
select Descricao, Titulo, TempoViagemRestanteMins, CounterValueBefore, Classificacao from dataoutput d where (Titulo = 4876 or Titulo = 4878 or Titulo = 4885 or Titulo = 4887) and (TempoViagemRestanteMins = 0 and CounterValueBefore = 0);

select Descricao, Titulo, TempoViagemRestanteMins, CounterValueBefore, Classificacao from dataoutput d where (Titulo = 4876 or Titulo = 4878 or Titulo = 4885 or Titulo = 4887) and (TempoViagemRestanteMins != 0 or CounterValueBefore != 0);
```

Código 10 - Avaliação das classificações (3ª Regra)

Observamos que, na primeira a classificação foi constantemente "Fraude 3" e Legítimo" na segunda. Esta análise abrangeu um total de 1000 registos no primeiro caso e aproximadamente 1400 na segunda. Com base nestes resultados, é possível afirmar que a regra está efetuando a classificação dos dados com precisão.

8.2. Quarta regra

A metodologia aplicada nesta regra é bastante similar à da regra anterior, com a principal diferença sendo a replicação do valor de “CounterValueBefore” na coluna “CounterValueAfter”, resultando em valores idênticos em ambas. Além disso, a coluna “TipoEvento” é constantemente atribuída ao valor “1”, representando uma entrada. Com esta configuração, espera-se que a regra categorize os dados com a classificação “Erro do Sistema 4”.

O código 11 em MySQL insere novos registos na tabela “dataoutput” com valores derivados de registos existentes, além de atualizar duas colunas (“Titulo” e “Descricao”) para categorizar e descrever esses registos de acordo com condições específicas.

O primeiro bloco insere até 1.000 novos registos na tabela “dataoutput” baseados em registos existentes.

- A consulta **SELECT** copia dados dos registos onde “Linha” está entre 1035005530 e 1035586691, inserindo até 1.000 desses registos de forma aleatória (**ORDER BY RAND()**).
- O valor para “Linha” e “Codigo” é calculado para ser um a mais que o maior valor atual na tabela, garantindo que cada novo registo tenha identificadores únicos.
- Algumas colunas, como “TipoEvento” (configurado para 1) e “CounterValueAfter” (igual a “CounterValueBefore”), são definidas com valores fixos ou derivados.

O segundo bloco tem como objetivo modificar o valor de “Titulo” com base na divisão do número de “Linha” por 3.

- O **CASE** define “Titulo” para cada registo:
  - “4876” para linhas divisíveis por 3.
  - “4878” para linhas onde o resto da divisão por 3 é 1.
  - “4887” para linhas onde o resto é 2.
  - “4885” para qualquer outro caso.

O terceiro bloco tem como objetivo atualizar a coluna “Descricao” com base nos valores específicos de “Titulo”.

- Cada valor de “Titulo” é associado a uma descrição:
  - 4876: "Pré-comprado Viagens - Adulto".
  - 4878: "Pré-comprado Grupo - Adulto".
  - 4887: "Pré-comprado Grupo - Criança".
  - Qualquer outro Titulo: "Pré-comprado Viagens - Criança".

```
INSERT INTO dataoutput (
 DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
 TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
 ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
```

```
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
  DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  1 AS TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueBefore AS CounterValueAfter, Titulo,
  GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
ORDER BY RAND()
LIMIT 1000;

update dataoutput set Titulo = case
   WHEN (Linha % 3) = 0 THEN 4876
   WHEN (Linha % 3) = 1 THEN 4878
   WHEN (Linha % 3) = 2 THEN 4887
   ELSE 4885
end;

update dataoutput set Descricao = case
   WHEN Titulo = 4876 THEN "Pré-comprado Viagens - Adulto"
   WHEN Titulo = 4878 THEN "Pré-comprado Grupo - Adulto"
   WHEN Titulo = 4887 THEN "Pré-comprado Grupo - Criança"
   ELSE "Pré-comprado Viagens - Criança"
end;
```

Código 11 - Inserção de novos dados (4ª Regra)

As duas consultas MySQL seguintes extraem registos específicos da tabela “dataoutput”, filtrando-os com base em valores nas colunas “Titulo”, “CounterValueBefore”, “CounterValueAfter” e “TipoEvento”.

A primeira *query* tem como objetivo selecionar registos onde "Titulo" seja um dos valores específicos (4876, 4878, 4885 ou 4887), onde "CounterValueBefore" é igual a "CounterValueAfter" e "TipoEvento" seja "1".

A segunda *query* é similar à primeira consulta, mas agora busca registos onde "CounterValueBefore" é diferente de "CounterValueAfter".

```
select Descricao, Titulo, CounterValueBefore, CounterValueAfter, TipoEvento, Classificacao from dataoutput
where (Titulo = 4876 or Titulo = 4878 or Titulo = 4885 or Titulo = 4887) and CounterValueBefore =
CounterValueAfter and TipoEvento = 1

select Descricao, Titulo, CounterValueBefore, CounterValueAfter, TipoEvento, Classificacao from dataoutput
where (Titulo = 4876 or Titulo = 4878 or Titulo = 4885 or Titulo = 4887) and CounterValueBefore !=
CounterValueAfter and TipoEvento = 1
```

Código 12 - Avaliação das classificações (4ª Regra)

Na fase de avaliação dos resultados, foi notado que a classificação foi constantemente dada como "Erro do Sistema 4" na primeira e como "Legítimo" na segunda. Esta análise abrangeu um total de 1000 registos e aproximadamente 1270 registos, respetivamente. Com base nestes resultados, pode-se afirmar que a regra está realizando a classificação dos dados corretamente.

8.3. Quinta regra

Para avaliar a quinta regra, planeamos mais uma vez copiar algumas linhas já existentes na tabela e, na mesma instância, realizar algumas alterações para que a validação viole a regra em questão. Essas modificações incluem ajustes na coluna "DataHora", de forma que os valores estejam dentro do intervalo entre as 00:00 e as 06:30. Adicionalmente, na coluna "Descricao", as descrições devem estar restritas a "Pré-comprado – Viagens - Criança" ou "Pré-comprado – Grupo – Criança", sendo esta escolha feita aleatoriamente. É importante ainda garantir que o valor na coluna "Titulo" corresponda à descrição selecionada, confirmando a continuidade da consistência dos dados para a avaliação da regra. Neste contexto, espera-se que a regra classifique estes dados como "Suspeita de Fraude 5".

O código 13 em MySQL insere novos registos na tabela "dataoutput", atribui valores de "Titulo" com base em condições, e define a descrição correspondente para cada "Titulo".

O primeiro bloco insere até 1.000 novos registos baseados em dados existentes, com uma nova data/hora gerada aleatoriamente.

- "DataHora" é calculado usando **NOW()** para a data atual e valores aleatórios para horas, minutos e segundos.
- O valor de "Linha" e "Codigo" é definido para ser o maior valor existente na tabela "dataoutput" incrementado por 1, garantindo que cada novo registo tenha identificadores únicos.

- A seleção é feita de registos onde “Linha” está entre 1035005530 e 1035586691, organizados aleatoriamente (**ORDER BY RAND()**) e limitados a 1.000 inserções.

O segundo bloco modifica a coluna “Titulo” com base na divisão do valor de “Linha” por 3.

- “Titulo” é atualizado de acordo com a condição:
  - “4887” para registos onde “Linha” é divisível por 3.
  - “4885” onde o resto da divisão de “Linha” por 3 é 1.
  - “3804” para qualquer outro valor de Linha.

O terceiro bloco atualiza a coluna “Descricao” com base nos valores de “Titulo”.

- Cada “Titulo” recebe uma descrição específica:
  - **4887**: "Pré-comprado Grupo - Criança".
  - **4885**: "Pré-comprado Viagens - Criança".
  - **Outros valores de “Titulo”**: "Pré-comprado Multiviagens Diário - Criança - Combinado HF ECO Interurbano - F7".

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  CONCAT(
    DATE_FORMAT(NOW(), '%Y-%m-%d'),
    ' ',
    LPAD(FLOOR(RAND() * 6), 2, '0'),
    ':',
    LPAD(FLOOR(RAND() * 31), 2, '0'),
    ':',
    LPAD(FLOOR(RAND() * 60), 2, '0')
  ) AS DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
  DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
```

```
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo,
  GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
ORDER BY RAND()
LIMIT 1000;

update dataoutput set Titulo = case
   WHEN (Linha % 3) = 0 THEN 4887
   WHEN (Linha % 3) = 1 THEN 4885
   ELSE 3804
end;

update dataoutput set Descricao = case
   WHEN Titulo = 4887 THEN "Pré-comprado Grupo - Criança"
   WHEN Titulo = 4885 THEN "Pré-comprado Viagens - Criança"
   ELSE "Pré-comprado Multiviagens Diário - Criança - Combinado HF ECO Interurbano - F7"
end;
```

Código 13 - Inserção de novos dados (5ª Regra)

As duas *queries* MySQL seguintes têm como objetivo recuperar registos da tabela "dataoutput", filtrando-os com base em condições específicas relacionadas ao "Titulo" e à hora registada em "DataHora".

A primeira consulta seleciona registos que têm um dos três valores específicos para a coluna "Titulo" (4887, 4885 ou 3804) e que foram registados entre a meia-noite (00:00:00) e às 6h30 da manhã (06:30:00).

- A cláusula **WHERE** aplica dois filtros:
  - O primeiro filtro (**titulo = 4887 OR titulo = 4885 OR titulo = 3804**) limita os resultados apenas aos registos que correspondem a um dos três títulos.
  - O segundo filtro **CAST(DataHora AS TIME) BETWEEN '00:00:00' AND '06:30:00'** extrai a parte do tempo de "DataHora" e garante que ela esteja dentro do intervalo definido. Isso permite que se considere apenas os registos que foram inseridos ou atualizados nesse período específico da madrugada.

A segunda consulta seleciona registos que têm um dos três valores específicos para a coluna "Titulo" (4887, 4885 ou 3804) e que foram registados entre às 6h30 e 23h59 da noite (até o final do dia).

- Assim como na primeira consulta, a cláusula **WHERE** aplica dois filtros:
    - O primeiro filtro é o mesmo, limitando os resultados a títulos específicos.
    - O segundo filtro **CAST(DataHora AS TIME) BETWEEN '06:30:01' AND '23:59:59'** assegura que a parte do tempo de "DataHora" esteja no intervalo da tarde e noite, começando um segundo após as 6h30 até a última fração do dia.

```
SELECT DataHora, Descricao, Titulo, Classificacao
FROM dataoutput
WHERE (titulo = 4887 OR titulo = 4885 OR titulo = 3804)
  AND CAST(DataHora AS TIME) BETWEEN '00:00:00' AND '06:30:00';

SELECT DataHora, Descricao, Titulo, Classificacao
FROM dataoutput
WHERE (titulo = 4887 OR titulo = 4885 OR titulo = 3804)
  AND CAST(DataHora AS TIME) BETWEEN '06:30:01' AND '23:59:59';
```

Código 14 - Avaliação das classificações (5ª Regra)

Concluída a classificação das validações, constatamos que, na primeira *query* a classificação foi sempre "Suspeita de Fraude 5", enquanto na segunda foi sempre "Legítima". Esta análise abrangeu um total de 1000 registos no primeiro caso e aproximadamente 8 registos no segundo. Com base nestes resultados, embora o pequeno conjunto de validações na segunda *query*, pode-se concluir que a regra está realizando a classificação de acordo com desejado.

8.4. Sétima regra

Para avaliar a sétima regra, podemos adotar uma abordagem semelhante à utilizada na terceira condição da sexta regra. Neste sentido, consiste em alterar os valores da coluna "NumMeses" da tabela, atribuindo-lhes um valor exageradamente alto e irrealista, garantindo assim que não coincida com o número correto de meses para os títulos aqui analisados. O propósito é estabelecer um cenário no qual alguns títulos apresentem um número incorreto de meses. Ao criar este cenário e considerando a condição estabelecida para esta regra, espera-se que ela classifique estes casos como "Fraude / Erro do Sistema 7".

O código 15 em MySQL tem como objetivo alterar registos na tabela dataoutput.

- **UPDATE dataoutput**: O comando **UPDATE** indica que a tabela "dataoutput" será modificada.
- **SET NumMeses = 1000000**: Esta parte do comando especifica que a coluna "NumMeses" será atualizada para o valor 1000000 para os registos que atendem à condição definida na cláusula **WHERE**.
- **WHERE Titulo IN (...)**: A cláusula **WHERE** filtra quais registos devem ser atualizados. O uso de **IN** permite verificar se o valor da coluna "Titulo" está presente na lista de valores especificados entre parênteses. Neste caso, apenas os registos cuja coluna "Titulo" contém um dos números listados (por exemplo, 7, 24, 26, etc.) serão afetados pela atualização.

- **LIMIT 1500**: A cláusula **LIMIT** é utilizada para restringir a quantidade de registos a serem atualizados. Isso significa que, mesmo que existam mais de 1500 registos que atendam à condição na cláusula **WHERE**, apenas os primeiros 1500 serão alterados.

```
UPDATE dataoutput SET NumMeses = 1000000 WHERE Titulo IN (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857) limit 1500;
```

Código 15 - Modificação dos dados existentes (7ª Regra)

O código 16 em MySQL mostra duas *queries* que têm como objetivo recuperar informações da tabela "dataoutput", filtrando os resultados com base em duas condições diferentes para a coluna "NumMeses".

- **SELECT Descricao, Titulo, NumMeses, Classificacao**: Esta parte indica que as colunas "Descricao", "Titulo", "NumMeses" e "Classificacao" da tabela "dataoutput" serão selecionadas e exibidas no resultado da consulta.
- **FROM dataoutput**: Define a tabela de onde os dados serão extraídos, que neste caso é "dataoutput".
- **WHERE NumMeses > 12 AND (Titulo IN (...))**: Para a primeira consulta, esta cláusula **WHERE** filtra os registos que têm o valor da coluna NumMeses maior que 12 e o valor da coluna "Titulo" deve estar entre os números especificados na lista. A cláusula **IN** é usada para verificar se o "Titulo" está numa lista de valores.
- **WHERE NumMeses <= 12 AND (Titulo IN (...))**: A segunda consulta é semelhante à primeira, mas filtra os registros onde "NumMeses" é menor ou igual a 12, mantendo a mesma lista de valores para "Titulo".

```
select Descricao, Titulo, NumMeses, Classificacao from dataoutput where NumMeses > 12 and (Titulo IN (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857));

select Descricao, Titulo, NumMeses, Classificacao from dataoutput where NumMeses <= 12 and (Titulo IN (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857));
```

Código 16 - Avaliação das classificações (7ª Regra)

Com as classificações já atribuídas, foi verificado individualmente que todas as validações retornadas receberam uma classificação de "Fraude / Erro do Sistema 7" na primeira *query* e "Legítimo" na segunda. É relevante destacar que foram retornadas um total de 1500 e 1710 validações, respetivamente. Com base nestas observações, é possível concluir que a sétima regra está operando conforme o esperado, em conformidade com a sua conceção inicial.

8.5. Oitava regra

Para uma avaliação adequada da oitava regra, foi necessário adicionar novas validações à tabela. Por outras palavras, a tabela original continha apenas um tipo de título, tornando essencial a inclusão de mais validações para abranger todos os tipos de títulos. Este processo envolveu a adição de validações complementares, baseadas nas existentes, porém com algumas alterações. Estas edições incluíram a alteração dos valores na coluna "Titulo", a adaptação do texto na coluna "Descricao" de acordo com o número especificado na coluna anterior, bem como a atribuição de valores excessivamente elevados e irreais à coluna "NumDias" em algumas instâncias, cujo objetivo já foi discutido em regras anteriores. Por outro lado, o valor correto foi atribuído à mesma coluna de acordo com o título nas restantes validações.

Com este cenário estabelecido e seguindo a condição especificada na regra, espera-se que a regra classifique as validações com valores excessivamente altos na coluna "NumDias" como "Fraude / Erro do Sistema 8".

O código 17 em MySQL consiste em várias instruções que visam inserir e atualizar registos na tabela "dataoutput".

A primeira instrução tem como objetivo inserir novos registos na tabela "dataoutput", onde os dados são selecionados da própria tabela "dataoutput", especificamente dos registos cuja coluna "Linha" está entre 1035005530 e 1035586691. Para garantir que os novos registos não colidam com os existentes, para cada nova linha inserida, "Linha" e "Codigo" são gerados incrementando o maior valor atual dessas colunas. Por fim, o número máximo de registos a serem inseridos é limitado a 3000.

A segunda instrução tem como objetivo atualizar a coluna "Titulo" da tabela "dataoutput". O novo valor de "Titulo" é determinado com base no resto da divisão de "Linha" por 3:

- Se Linha % 3 = 0, o título é definido como 4823.
- Se Linha % 3 = 1, o título é definido como 5282.
- Caso contrário, o título é definido como 4829.

O papel da terceira instrução é atualizar a coluna "Descricao" com valores específicos com base nos valores de "Titulo":

- Se **Titulo = 4823**: "Passe Social - 15 dias"
- Se **Titulo = 5282**: "Passe Férias Estudante - 15 dias"
- Se **Titulo = 4829**: "Passe Social - 30 dias"

A quarta instrução tem como objetivo atualizar o campo "NumDias" para o valor 1.000.000 para os registos cujo "Titulo" esteja na lista fornecida. A cláusula **IN** é usada para verificar se o "Titulo" está numa lista de valores. Vale ressaltar que esta atualização é limitada a 2000 registos.

A quinta instrução tem a finalidade de atualizar o valor de “NumDias” para valores reais (15 ou 30) dependendo do “Titulo”, mas apenas para registos que não foram previamente definidos como 1.000.000. É utilizado uma estrutura CASE para determinar o número correto de dias com base no título.

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
  DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo,
  GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
LIMIT 3000;

UPDATE dataoutput
SET Titulo = CASE
    WHEN (Linha % 3) = 0 THEN 4823
    WHEN (Linha % 3) = 1 THEN 5282
    ELSE 4829
END;



update dataoutput set Descricao = case
            WHEN Titulo = 4823 then "Passe Social - 15 dias"
```

```
   WHEN Titulo = 5282 then "Passe Férias Estudante - 15 dias"
   when Titulo = 4829 then "Passe Social  - 30 dias"
end;

update dataoutput set NumDias = 1000000 where Titulo in (4823, 5282, 4829) limit 2000;

update dataoutput set NumDias = case
            when Titulo = 4823 then 15
            when Titulo = 5282 then 15
            when Titulo = 4829 then 30
end
where NumDias != 1000000;
```

Código 17 - Inserção de novos dados (8ª Regra)

O código 18 em MySQL contém duas instruções **SELECT** que visam consultar dados da tabela "dataoutput" com base em condições específicas sobre os campos "NumDias" e "Titulo".

A primeira consulta tem como objetivo seleciona registos da tabela "dataoutput" onde o número de dias ("NumDias") é maior que 30 e o título ("Titulo") é um dos três valores especificados (4823, 5282, 4829). Além disso, é importante mencionar que os campos retornados são "Descricao", "Titulo", "NumDias" e "Classificacao". O uso da cláusula **IN** permite verificar se o valor da coluna "Titulo" corresponde a qualquer um dos valores listados.

A segunda consulta tem um objetivo semelhante à primeira, mas aqui seleciona registos onde "NumDias" é menor ou igual a 30. Ela utiliza a mesma lógica com a cláusula **IN** para filtrar os registos de acordo com os valores do campo "Titulo". Adicionalmente, assim como na primeira consulta, os campos retornados são "Descricao", "Titulo", "NumDias" e "Classificacao".

```
select Descricao, Titulo, NumDias, Classificacao from dataoutput where NumDias > 30   and Titulo in (4823, 5282, 4829);

select Descricao, Titulo, NumDias, Classificacao from dataoutput where NumDias <= 30 and Titulo in (4823, 5282, 4829);
```

Código 18 - Avaliação das classificações (8ª Regra)

Verificamos que todas as 2000 validações na primeira *query* receberam a classificação "Fraude / Erro do Sistema 8", enquanto na segunda todas as 9000 validações foram classificadas como "Legítimo". Considerando estas observações, podemos afirmar que a oitava regra está classificando conforme o desejado.

8.6. Nona regra

Para avaliar a nona regra, foi necessário mais uma vez introduzir novas validações na tabela. Isto deve-se à escassez de validações em que a data e a hora são muito próximas à validação anterior para o mesmo cliente. Esta falta pode comprometer a avaliação da precisão das classificações da regra, por isso, a inclusão de novas que vão contra a condição estabelecida pela regra é importante para avaliar com precisão as classificações atribuídas por ela.

Uma abordagem direta para favorecer as condições da regra consiste em configurar a "DataHora" da validação como igual à coluna "UltimaValidacao" e também à coluna "UltimoCooldownEnd", enquanto se atribui na mesma instância o valor "1" à coluna "UltimaPosValid".

Com esta configuração e considerando a condição estipulada pela regra, espera-se que ela classifique estes casos como "Suspeita de Fraude 9".

O código 19 em MySQL tem como objetivo inserir dados na tabela "dataoutput", listando todas as colunas que receberão os novos valores.

- **SELECT (..)**: Seleciona os dados que serão inseridos na tabela "dataoutput".
    - **(SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha** e **(SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo**; Para as colunas "Linha" e "Codigo", o código busca o valor máximo atual de cada uma dessas colunas e adiciona 1 para gerar um novo identificador exclusivo para cada inserção.
    - **DataHora as UltimaValidacao**, **1 as UltimoPosValid**, **DataHora as UltimoCooldownEnd**: Para alguns campos, como "UltimaValidacao" e "UltimoCooldownEnd", o código utiliza o valor da coluna "DataHora" como referência, enquanto "UltimoPosValid" é definido como 1.
- **WHERE Linha BETWEEN 1035005530 AND 1035586691**: Especifica que apenas as linhas da tabela "dataoutput" que possuem um valor de "Linha" entre 1035005530 e 1035586691 serão consideradas para a inserção.
- **LIMIT 3000**: Restringe o número de registos selecionados para um máximo de 3000, evitando que a inserção exceda esse número.

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
```

```
 DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
 DataHoraRegisto,
 (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
 (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
 TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
 ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
 CounterValueBefore, CounterValueAfter, Titulo,
 GrupoTitulo, NumViagens, NumDias, NumMeses,
 TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, DataHora as UltimaValidacao,
 CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, 1 as UltimoPosValid, DataHora as UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
LIMIT 3000;
```

Código 19 - Inserção de novos dados (9ª Regra)

O código 20 em MySQL contém duas consultas que selecionam dados de uma tabela chamada dataoutput.

A primeira seleciona colunas específicas (“Descricao”, “Titulo”, “DataHora”, “UltimaValidacao”, “UltimoCooldownEnd”, “UltimoPosValid”, “Classificacao”) da tabela “dataoutput”, no qual tem as seguintes condições:

- **DataHora >= UltimaValidacao**: Filtra os registos em que “DataHora” é maior ou igual à “UltimaValidacao”, ou seja, a data/hora do registo deve ser após ou igual à última validação.
- **DataHora <= UltimoCooldownEnd**: Garante que “DataHora” não ultrapasse “UltimoCooldownEnd”, ou seja, os registos devem estar dentro do período de *cooldown*.
- **UltimoPosValid = 1**: Filtra registos onde a coluna “UltimoPosValid” é igual a 1, indicando que a validação foi válida.

Já a segunda, assim como a primeira consulta, esta seleciona as mesmas colunas da tabela “dataoutput”. As condições são as seguintes:

- **DataHora > UltimaValidacao**: Aqui, os registos devem ter uma “DataHora” que é estritamente posterior à “UltimaValidacao”.
- **DataHora > UltimoCooldownEnd**: A “DataHora” também deve ser posterior ao “UltimoCooldownEnd”, indicando que esses registos ocorreram após o fim do período de *cooldown*.

```
select Descricao, Titulo, DataHora, UltimaValidacao, UltimoCooldownEnd, UltimoPosValid, Classificacao from
dataoutput where DataHora >= UltimaValidacao and DataHora <= UltimoCooldownEnd and UltimoPosValid = 1
```

```
select Descricao, Titulo, DataHora, UltimaValidacao, UltimoCooldownEnd, UltimoPosValid, Classificacao from dataoutput where DataHora > UltimaValidacao and DataHora > UltimoCooldownEnd
```

Código 20 - Avaliação das classificações (9ª Regra)

Observamos que a primeira *query* classificou todas as validações como "Suspeita de Fraude 9", e a segunda classificou todas como "Legítimo". É relevante destacar que a primeira retornou um total de 1864 registos, enquanto a segunda um total de 2650 registos. Com base nestes resultados, é possível concluir que a nona regra está operando adequadamente como desejado.

8.7. Décima regra

A avaliação da décima regra será realizada da mesma maneira do que a sexta regra, sendo dividida em três partes cujo objetivo já foi discutido anteriormente.

Primeira condição

Pata avaliar a primeira condição da décima regra, não foi necessário realizar alterações nas validações existentes, nem inserir novas, pois já há um número suficiente onde "DHIniViagem" está como "0000-00-00 00:00:00" e que não seja bilhetes de bordo. Desta forma, é necessário executar as *queries* abaixo para exibirem todas as validações, que por sua vez serão avaliadas. As validações que cumprem com a condição mencionada anteriormente deverão estar classificadas como "Dados em Falta", enquanto as demais como "Legítimo".

O código 21 em MySQL consiste em duas consultas distintas que selecionam dados da tabela "dataoutput".

A primeira consulta tem como objetivo seleciona registos únicos (distintos) das colunas "DHIniViagem", "Descricao" e "Classificacao" da tabela "dataoutput". As suas condições são as seguintes:

- **DHIniViagem = "0000-00-00 00:00:00"**: Filtra os registos em que a coluna "DHIniViagem" possui o valor "0000-00-00 00:00:00", que geralmente indica uma data/hora inválida ou não especificada.
- **Descricao != "Bilhete de Bordo"**: Exclui os registos cuja descrição é "Bilhete de Bordo", ou seja, esses registos não são relevantes para o resultado.

A segunda consulta novamente, seleciona registos únicos das mesmas colunas ("DHIniViagem", "Descricao", "Classificacao") da tabela "dataoutput", onde as suas condições são as seguintes:

- **DHIniViagem != "0000-00-00 00:00:00"**: Filtra os registos onde a coluna "DHIniViagem" tem um valor diferente de "0000-00-00 00:00:00", ou seja, aqui, apenas registros com datas/hora válidas serão considerados.
- **Descricao != "Bilhete de Bordo"**: Assim como na primeira consulta, essa condição exclui registos com a descrição "Bilhete de Bordo".

```
select distinct DHIniViagem, Descricao, Classificacao from dataoutput d where DHIniViagem = "0000-00-00 00:00:00" and Descricao != "Bilhete de Bordo"

select distinct DHIniViagem, Descricao, Classificacao from dataoutput d where DHIniViagem != "0000-00-00 00:00:00" and Descricao != "Bilhete de Bordo"
```

Código 21 - Avaliação das classificações (10ª Regra – 1ª Condição)

Concluída a análise dos dados, observamos que a primeira classificou todas as 59 validações como “Dados em Falta”, enquanto a segunda classificou as 14115 validações como “Legítimo”, lembrando que ambas estão aplicando o modificador “Distinct” para apresentar as validações divergentes. Com base nestes resultados, podemos concluir que a primeira condição da décima regra está funcionando conforme esperado.

Segunda condição

Para avaliar a segunda condição da décima regra, não foi necessário novamente realizar modificações ou inserções, pois já há um número suficiente de registos onde o campo “DHIniViagem” não é “0000-00-00 00:00:00”, enquanto o campo “UltimoDHIniViagem é “0000-00-00 00:00:00”. Além disso, estas validações não devem corresponder à primeira validação do cliente e não podem ser bilhetes de bordo. Desta forma, as validações que cumprem com as estas condição deverão estar classificadas como “Inválido”, por outro lado, devem ser consideradas como “Legítimo”. Finalmente, é necessário executar as *queries* seguintes de forma a exibirem todas as validações para tornar mais viável a avaliação.

O código 22 em MySQL possuí três consultas que selecionam registos únicos de algumas colunas (“DHIniViagem”, “Descricao”, “Classificacao”) da tabela “dataoutput” com base em condições específicas aplicadas sobre colunas de data e descrição.

A primeira consulta tem como objetivo selecionar registos distintos das colunas “DHIniViagem”, “Descricao”, e “Classificacao” da tabela “dataoutput”, que por sua vez tem as seguintes condições:

- **DHIniViagem != "0000-00-00 00:00:00"**: Considera apenas os registos onde “DHIniViagem” tem uma data válida (diferente de "0000-00-00 00:00:00").
- **UltimoDHIniViagem = "0000-00-00 00:00:00"**: Garante que “UltimoDHIniViagem” está vazio ou inválido, ou seja, tem o valor "0000-00-00 00:00:00".
- **DataHora != PrimeiraValidacao**: Inclui apenas os registos onde “DataHora” é diferente de “PrimeiraValidacao”, ou seja, não é a primeira validação do cliente.
- **Descricao != "Bilhete de Bordo"**: Exclui registos com a descrição "Bilhete de Bordo".

A segunda consulta tem um objetivo semelhante à primeira, onde seleciona registos distintos para as mesmas colunas. As suas condições são as seguintes:

- **DHIniViagem != "0000-00-00 00:00:00"**: Considera apenas os registos onde “DHIniViagem” tem uma data válida.

- **UltimoDHIniViagem = "0000-00-00 00:00:00"**: Apenas os registos onde "UltimoDHIniViagem" está vazio ou inválido são incluídos.
- **DataHora = PrimeiraValidacao**: Inclui apenas os registos onde "DataHora" é igual a "PrimeiraValidacao", ou seja, é a primeira validação do cliente.
- **Descricao != "Bilhete de Bordo"**: Exclui os registos com a descrição "Bilhete de Bordo".

A terceira consulta, novamente, seleciona registos distintos para as mesmas colunas, onde as suas condições são as seguintes:

- **DHIniViagem != "0000-00-00 00:00:00"**: Apenas os registos onde "DHIniViagem" tem uma data válida são incluídos.
- **UltimoDHIniViagem != "0000-00-00 00:00:00"**: Neste caso, "UltimoDHIniViagem" também precisa ter uma data válida.
- **DataHora != PrimeiraValidacao**: Exclui os registros onde "DataHora" é igual a "PrimeiraValidacao", incluindo apenas a primeira validação do cliente.
- **Descricao != "Bilhete de Bordo"**: Exclui novamente os registos com a descrição "Bilhete de Bordo".

```
select distinct DHIniViagem, Descricao, Classificacao from dataoutput d where DHIniViagem != "0000-00-00 00:00:00" and UltimoDHIniViagem = "0000-00-00 00:00:00" and DataHora != PrimeiraValidacao and Descricao != "Bilhete de Bordo"

select distinct DHIniViagem, Descricao, Classificacao from dataoutput d where DHIniViagem != "0000-00-00 00:00:00" and UltimoDHIniViagem = "0000-00-00 00:00:00" and DataHora = PrimeiraValidacao and Descricao != "Bilhete de Bordo"

select distinct DHIniViagem, Descricao, Classificacao from dataoutput d where DHIniViagem != "0000-00-00 00:00:00" and UltimoDHIniViagem != "0000-00-00 00:00:00" and DataHora != PrimeiraValidacao and Descricao != "Bilhete de Bordo"
```

Código 22 - Avaliação das classificações (10ª Regra – 2ª Condição)

Após executar as *queries*, notamos que a primeira classificou todas as validações como "Inválido", quanto a segunda e a terceira as consideraram "Legítimo". É importante mencionar que a primeira consulta retornou 3700 validações, a segunda 6250 e a terceira 7136, todas usando o modificador "Distinct". Com base nesses resultados, podemos concluir que a segunda condição da décima está funcionando conforme o esperado.

Terceira condição

Para avaliar a terceira condição da décima regra, podemos inserir novos dados e, na mesma instância, realizar ajustes nas validações existentes, especificamente na coluna "Veiculo" para coincidir com os valores da coluna "UltimoVeiculo", na coluna "DHIniViagem" para corresponder ao valor da coluna "UltimoDHIniViagem", e na coluna "UltimoPosValid" para atribuir o valor "1". O propósito é criar um cenário no qual existe uma segunda

validação para o mesmo cliente, veículo e viagem, considerando que a sua validação prévia foi aprovada. Com este cenário estabelecido e em conformidade com o que foi mencionado, espera-se que tais casos sejam classificados como "Fraude 10".

O código 23 em MySQL é uma instrução para inserir novos dados em uma tabela chamada dataoutput, utilizando informações provenientes da própria tabela.

A instrução **INSERT INTO** é usada para adicionar novas linhas à tabela "dataoutput". Ela lista todas as colunas da tabela que receberão os valores novos. Estas colunas incluem informações como "DataHora", "Carreira", "Chapa", "DHIniViagem", entre muitas outras.

O **SELECT** pega os dados já existentes na tabela "dataoutput" para preencher as colunas especificadas. Cada coluna recebe os valores já existentes, com exceção das colunas "Linha" e "Codigo" que recebem valores calculados, e algumas outras colunas que possuem valores específicos. Para as colunas "Linha" e "Codigo", há uma subconsulta que utiliza **MAX(Linha)** e **MAX(Codigo)**, ou seja, o maior valor presente nas colunas aqui em causa, e soma 1 a elas. Isto cria um novo valor para essas colunas, evitando duplicidades. A coluna "UltimoVeiculo" recebe o valor da coluna "Veiculo" atual. Da mesma forma, "UltimoDHIniViagem" e "UltimoPosValid" recebem valores fixos ("DHIniViagem" e 1, respectivamente).

A condição **WHERE** especifica que somente as linhas onde a coluna "Linha" está entre os valores 1035005530 e 1035586691 serão usadas na operação, limitando os dados que serão copiados para os novos registos.

Por fim, **LIMIT 400** restringe o número de registos que a operação de inserção afetará, garantindo que apenas 400 registos (no máximo) sejam inseridos.

```
INSERT INTO dataoutput (
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao, DataHoraRegisto, Linha, Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
  CounterValueBefore, CounterValueAfter, Titulo, GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, UltimoVeiculo, UltimoDHIniViagem, UltimoPosValid, UltimoCooldownEnd
)
SELECT
  DataHora, Carreira, Chapa, DHIniViagem, Activo, Descricao,
  DataHoraRegisto,
  (SELECT MAX(Linha) FROM dataoutput) + 1 AS Linha,
  (SELECT MAX(Codigo) FROM dataoutput) + 1 AS Codigo,
  TipoEvento, Veiculo, PosValid, VarPercurso, SentidoPerc, Viagem, ParagINIOrd, ParagFIMOrd,
  ParagemEntradaOrd, TempoViagemRestanteMins, ZonaCorrente, NSCartaoHI, NSCartaoLO,
```

```
  CounterValueBefore, CounterValueAfter, Titulo,
  GrupoTitulo, NumViagens, NumDias, NumMeses,
  TipoTarifa, Operador, Rede, NSCartao, Classificacao, PrimeiraValidacao, UltimaValidacao,
  CooldownEnd, Veiculo as UltimoVeiculo, DHIniViagem as UltimoDHIniViagem, 1 as UltimoPosValid,
UltimoCooldownEnd
FROM dataoutput
WHERE Linha BETWEEN 1035005530 AND 1035586691
LIMIT 400;
```

Código 23 - Inserção de novos dados (10ª Regra – 3ª Condição)

O código 24 em MySQL tem duas instruções que extraem dados específicos da tabela "dataoutput", com o objetivo de analisar os registos de viagem e os detalhes associados a cada veículo.

A primeira consulta começa selecionando as colunas "DataHora", "NSCartao", "Veiculo", "UltimoVeiculo", "DHIniViagem", "UltimoDHIniViagem", "UltimoPosValid" e "Classificacao" da tabela "dataoutput". Em seguida, a instrução aplica várias condições de filtragem para garantir que apenas os registos que satisfaçam certas condições sejam incluídos no resultado. Primeiramente, é verificado se as colunas "DHIniViagem" e "UltimoDHIniViagem" contêm datas válidas, excluindo os registos que possuam valores "0000-00-00 00:00:00", o que representam uma data inválida. Depois disso, a condição **Veiculo = UltimoVeiculo** verifica se o veículo atual ("Veiculo") é o mesmo que o último veículo registrado ("UltimoVeiculo"). A condição seguinte, **DHIniViagem = UltimoDHIniViagem**, assegura que o horário de início da viagem atual é igual ao horário de início da última viagem registada, o que indica continuidade ou repetição de uma viagem anterior no mesmo horário. Finalmente, **UltimoPosValid = 1** confirma que a última validação foi validada.

A segunda consulta também seleciona as colunas "DataHora", "NSCartao", "Veiculo", "UltimoVeiculo", "DHIniViagem", "UltimoDHIniViagem", "UltimoPosValid" e "Classificacao" da mesma tabela "dataoutput". No entanto, ela busca um conjunto diferente de registos ao aplicar condições que indicam mudanças entre o registo atual e o último registo armazenado. As duas primeiras condições, **DHIniViagem != "0000-00-00 00:00:00"** e **UltimoDHIniViagem != "0000-00-00 00:00:00"**, permanecem as mesmas da consulta anterior e garantem que ambos os horários de início de viagem sejam válidos. A próxima condição, **Veiculo != UltimoVeiculo**, identifica uma mudança no veículo entre o registro atual e o último registro, mostrando que o veículo atual é diferente do veículo registado anteriormente. Em seguida, **DHIniViagem != UltimoDHIniViagem** verifica se o horário de início da viagem atual é diferente do horário de início da viagem anterior, indicando que houve uma alteração no horário.

```
select DataHora, NSCartao, Veiculo, UltimoVeiculo, DHIniViagem, UltimoDHIniViagem, UltimoPosValid,
Classificacao from dataoutput where DHIniViagem != "0000-00-00 00:00:00" and UltimoDHIniViagem !=
"0000-00-00 00:00:00" and Veiculo = UltimoVeiculo and DHIniViagem = UltimoDHIniViagem and
UltimoPosValid = 1
```

```
select DataHora, NSCartao, Veiculo, UltimoVeiculo, DHIniViagem, UltimoDHIniViagem, UltimoPosValid,
Classificacao from dataoutput where DHIniViagem != "0000-00-00 00:00:00" and UltimoDHIniViagem !=
"0000-00-00 00:00:00" and Veiculo != UltimoVeiculo and DHIniViagem != UltimoDHIniViagem
```

Código 24 - Avaliação das classificações (10ª Regra – 3ª Condição)

Durante a análise, notamos que a primeira classificou todas as 376 validações como "Fraude 10", enquanto a segunda considerou as 293 validações como "Legítimas". Assim sendo, com base nestes resultados, podemos concluir que a terceira condição da décima regra está funcionando conforme esperado.

8.8. Décima primeira regra

Para verificar a eficácia da décima primeira regra, podemos realizar algumas modificações nas validações existentes, focando especificamente na coluna "Operador", onde o valor "5" é atribuído para representar o operador interurbano, e na coluna "Rede", onde os valores "256", "257" e "259" correspondem às três redes interurbanas. O objetivo é simular um cenário no qual um título urbano seja validado num operador ou rede interurbana, ou vice-versa, por exemplo. Com este cenário estabelecido e tendo em conta as condições definidas nesta regra, espera-se que estes casos sejam classificados como "Fraude / Erro do Sistema 11".

O código 25 em MySQL é usada para modificar os valores de algumas colunas em registos específicos da tabela dataoutput.

A instrução começa com **UPDATE dataoutput SET**, indicando que os registos na tabela "dataoutput" serão atualizados, onde dois campos serão alterados: Operador e Rede. A coluna "Operador" é definida para todos os registos serem alterados para o 5, desde que o registo que atenda às condições especificadas no final da instrução. A coluna "Rede" é atualizada com valores diferentes, dependendo de uma condição baseada no valor da coluna "Linha". Para isso, é usada uma instrução CASE:

- Quando Linha % 3 = 0, ou seja, quando Linha é divisível por 3, o valor 256 é atribuído à coluna "Rede".
- Quando Linha % 3 = 1, ou seja, quando a divisão de Linha por 3 tem resto 1, o valor 257 é atribuído à coluna "Rede".
- Em qualquer outro caso, o valor 259 é atribuído à coluna "Rede".

A instrução contínua com a condição **WHERE** que especifica que apenas os registos onde a coluna "" contém um dos valores listados (por exemplo, 4820, 5076, 4858, etc.) serão atualizados. Esse filtro é uma lista de números específicos na coluna "Titulo", e somente os registos que têm um desses valores passarão pela atualização. Por fim, a cláusula **LIMIT 1000** define que, no máximo, 1000 registos serão atualizados, mesmo que haja mais registos que atendam aos critérios da condição no **WHERE**.

```
update dataoutput set Operador = 5, Rede = CASE
  WHEN (Linha % 3) = 0 THEN 256
  WHEN (Linha % 3) = 1 THEN 257
```

```
   ELSE 259
end
where Titulo in (4820, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9031, 9045, 9059, 9073, 4857, 4823, 5282, 4829)
limit 1000;
```

Código 25 - Modificação dos dados existentes (11ª Regra)

No código 26 em MySQL tem duas instruções **SELECT** que consultam a tabela "dataoutput" para retornar dados específicos sobre os registos que possuem certos valores na coluna "Titulo". Ambas as consultas focam nos campos "Descricao", "Titulo", "Operador", "Rede" e "Classificacao", mas aplicam condições diferentes em relação aos valores das colunas "Operador" e "Rede".

A primeira consulta seleciona as colunas já mencionadas da tabela "dataoutput". Em seguida, aplica uma condição **WHERE** para filtrar os registos de acordo com duas restrições principais:

- A primeira parte da condição **WHERE** especifica que a coluna "Titulo" deve conter um dos valores fornecidos na lista, como 4820, 5076, 4858, entre outros. Isso garante que apenas registos com esses valores em "Titulo" serão considerados.
- A segunda parte da condição é composta por uma combinação de restrições em "Operador" e "Rede". Ela utiliza a cláusula (**Operador = 5 OR (Rede = 256 OR Rede = 257 OR Rede = 259)**), o que significa que, para que o registo seja incluído no resultado, ele deve atender a pelo menos uma das seguintes condições:
    - O valor de Operador deve ser igual a 5.
    - Ou então, Rede deve ser igual a 256, 257 ou 259.

A segunda consulta é semelhante na estrutura, mas com uma diferença importante nas condições da cláusula **WHERE**:

- Assim como na primeira consulta, a segunda consulta aplica a condição para que "Titulo" esteja na mesma lista de valores, restringindo a consulta aos mesmos registos.
- A segunda parte da condição **WHERE** especifica que, para um registo ser incluído, ele deve atender às três condições seguintes:
    - Operador deve ser diferente de 5.
    - Rede deve ser diferente de 256.
    - Rede deve ser diferente de 257.
    - Rede deve ser diferente de 259.

```
select Descricao, Titulo, Operador, Rede, Classificacao from dataoutput where Titulo in (4820, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9031, 9045, 9059, 9073, 4857, 4823, 5282, 4829) and (Operador = 5 or (Rede = 256 or Rede = 257 or Rede = 259))
```

```
SELECT Descricao, Titulo, Operador, Rede, Classificacao
FROM dataoutput
WHERE Titulo IN (4820, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 4837, 4839, 4841, 4843, 4849, 4854,
5283, 9003, 9017, 9031, 9045, 9059, 9073, 4857, 4823, 5282, 4829)
AND (Operador != 5 AND Rede != 256 AND Rede != 257 AND Rede != 259)
```

Código 26 - Avaliação das classificações (11ª Regra)

Com as classificações reveladas, observou-se que a primeira categorizou todas as validações como "Fraude / Erro do Sistema 11", enquanto a segunda classificou todas como "Legítimo". É relevante notar que a primeira *query* retornou 1500 validações, enquanto a segunda retornou 1860 validações. Com base nestes resultados, podemos inferir que a décima segunda regra está classificando as validações conforme o esperado.

8.9. Décima segunda regra

Para avaliar a eficácia da décima segunda regra, foram realizadas modificações nas validações existentes, particularmente na coluna "TipoTarifa". Estas alterações envolveram a modificação dos valores de alguns títulos e em algumas validações, introduzindo tarifas incorretas para o tipo de título correspondente. Por exemplo, se um título estava associado à tarifa 1, o seu valor foi alterado para 2 ou 3; da mesma forma, se um título estava associado à tarifa 2, o seu valor foi alterado para 1 ou 3, e assim por diante para os títulos associados à tarifa 3. Este processo tem como objetivo simular cenários nos quais os títulos validados não possuem a tarifa correta para o seu tipo. Com este cenário estabelecido e considerando as condições definidas na regra, espera-se que estes casos aqui descritos sejam classificados como "Fraude / Erro do sistema 12.1" e/ou "Fraude / Erro do sistema 12.2" e/ou "Fraude / Erro do sistema 12.3", respetivamente.

O código 27 em MySQL possui três instruções **UPDATE** ajustam a coluna "TipoTarifa" na tabela "dataoutput", aplicando uma atualização condicional para registos específicos baseados nos valores da coluna "Titulo". Cada instrução utiliza condições com **CASE** para definir o valor de "TipoTarifa" conforme o valor da coluna "Linha", que é o identificador da linha do registo.

Na primeira instrução a condição **WHERE** restringe os registos para atualização apenas àqueles onde "Titulo" tem um dos valores 4885, 4887 ou 3804. Em seguida, a instrução **SET TipoTarifa = CASE** usa uma lógica condicional:

- Se o valor de Linha % 3 é igual a 0, então "TipoTarifa" é definido como 2.
- Se Linha % 3 é igual a 1, então "TipoTarifa" é definido como 3.
- Para qualquer outro caso, "TipoTarifa" mantém seu valor atual.

Adicionalmente, a cláusula **LIMIT 8** limita essa atualização a, no máximo, 8 registos que atendem a essas condições.

Na segunda instrução os registos são filtrados, igualmente, com **WHERE Titulo IN (...)**, restringindo-se a valores de Titulo específicos como 9031, 9032, 9034, entre outros. A lógica para "TipoTarifa" é semelhante, com alterações nos valores atribuídos:

- Quando Linha % 3 = 0, "TipoTarifa" é atualizado para 1.
- Quando Linha % 3 = 1, "TipoTarifa" é atualizado para 3.
- Nos demais casos, "TipoTarifa" mantém seu valor atual.

Por fim, neste caso, o limite é de 715 registos, ou seja, no máximo 715 registos que atendem às condições serão atualizados.

```
update dataoutput set TipoTarifa = case
   WHEN (Linha % 3) = 0 THEN 2
   WHEN (Linha % 3) = 1 THEN 3
   else TipoTarifa
end
where Titulo in (4885, 4887, 3804)
limit 8;

update dataoutput set TipoTarifa = case
            when (Linha % 3) = 0 then 1
            when (Linha % 3) = 1 then 3
            else TipoTarifa
end
where Titulo in (9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058)
limit 715;

update dataoutput set TipoTarifa = case
            when (Linha % 3) = 0 then 1
            when (Linha % 3) = 1 then 2
            else TipoTarifa
end
where Titulo in (69, 71, 74, 75, 4486, 4854, 9059, 9060, 9065, 9072, 9073, 9074, 9079)
limit 588;
```

Código 27 - Modificação dos dados existentes (12ª Regra)

A avaliação desta regra seguirá a metodologia da sexta regra, onde ela é simplificada pela abordagem de análise por condição de forma individual.

Primeira condição

O código 28 em MySQL contém duas consultas **SELECT** que extraem dados da tabela "dataoutput" com o objetivo de analisar registos onde a coluna "Titulo" possui valores específicos e verificar quais deles possuem o valor 1 em "TipoTarifa" e quais não possuem. Ambas as consultas selecionam as colunas "Descricao", "Titulo", "TipoTarifa" e "Classificacao", mas aplicam filtros diferentes para "TipoTarifa".

Na primeira consulta, a condição **WHERE Titulo IN (4885, 4887, 3804)** seleciona apenas os registos em que "Titulo" é um dos valores específicos da lista: 4885, 4887, ou 3804. Além disso, a condição **AND TipoTarifa = 1** filtra os registos para aqueles onde "TipoTarifa" é igual a 1.

Na segunda consulta, a condição **WHERE Titulo IN (4885, 4887, 3804)** permanece a mesma. Já a condição **AND TipoTarifa != 1** filtra para os registos onde "TipoTarifa" não é igual a 1.

```
select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (4885, 4887, 3804) and TipoTarifa = 1

select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (4885, 4887, 3804) and TipoTarifa != 1
```

Código 28 - Avaliação das classificações (12ª Regra – 1ª Condição)

Durante a análise, foi observado que a primeira retornou todas as validações como "Legítimo", enquanto a segunda retornou todas como "Fraude / Erro do sistema 12.1". É importante notar que a primeira *query* apresentou 3 validações, enquanto a segunda, apresentou 5 validações. Embora seja um conjunto de dados pequeno, a precisão na classificação de todas as validações pela regra sugere que não devem surgir problemas ao lidar com conjuntos de dados mais extensos. Com base nestes resultados, pode-se concluir que a primeira condição da décima segunda regra está apta a classificar as validações para este tipo específico de tarifa.

Segunda condição

O código 29 em MySQL, semelhantemente à anterior, é composta por duas instruções **SELECT** que têm como objetivo consultar a tabela "dataoutput", especificamente buscando registos que têm valores especificos na coluna "Titulo" e verificando o valor da coluna "TipoTarifa". Ambas as consultas selecionam as colunas "Descricao", "Titulo", "TipoTarifa" e "Classificacao", mas aplicam filtros diferentes em relação ao campo "TipoTarifa".

Na primeira *query* a condição **WHERE Titulo IN (9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058)** restringe os resultados aos registos em que a coluna "Titulo" contém um dos valores listados (9031, 9032, 9034, etc.). Já a condição adicional **AND TipoTarifa = 2** filtra ainda mais os resultados para incluir apenas aqueles onde "TipoTarifa" é igual a 2.

Relativamente à segunda *query*, a condição **WHERE Titulo IN (9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058)** permanece a mesma. Por outro lado, a condição **AND TipoTarifa != 2** filtra os resultados para incluir apenas os registos onde "TipoTarifa" não é igual a 2.

```
select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058) and TipoTarifa = 2

select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058) and TipoTarifa != 2
```

Código 29 - Avaliação das classificações (12ª Regra – 2ª Condição)

Após a execução das *queries* anteriores, constatou-se que a primeira retornou todas as 239 validações como "Legítimo", enquanto a segunda retornou todas as 476 como "Fraude / Erro do sistema 12.2". A partir destes resultados, é possível concluir que a segunda condição desta regra está efetuando a classificação das validações corretamente para este tipo específico de tarifa.

Terceira condição

O código 30 em MySQL é composto por duas instruções **SELECT** que são projetadas para consultar a tabela "dataoutput", assim como nas consultas anteriores, focando em registos com valores específicos na coluna "Titulo" e analisando o valor da coluna "TipoTarifa". Cada consulta seleciona as mesmas colunas que as anteriormente discutidas.

A primeira consulta contém a condição **WHERE Titulo IN (69, 71, 74, 75, 4486, 4854, 9059, 9060, 9065, 9072, 9073, 9074, 9079)** que limita os resultados a registos com títulos específicos. A mesma ainda possui uma segunda condição que restringe ainda mais a seleção para incluir apenas aqueles registos onde "TipoTarifa" é igual a 3 (**AND TipoTarifa = 3**).

A diferença da segunda consulta para a primeira reside na condição **AND TipoTarifa != 3** que inclui apenas aqueles registros onde "TipoTarifa" não é igual a 3.

```
select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (69, 71, 74, 75, 4486, 4854, 9059, 9060, 9065, 9072, 9073, 9074, 9079) and TipoTarifa = 3

select Descricao, Titulo, TipoTarifa, Classificacao from dataoutput where Titulo in (69, 71, 74, 75, 4486, 4854, 9059, 9060, 9065, 9072, 9073, 9074, 9079) and TipoTarifa != 3
```

Código 30 - Avaliação das classificações (12ª Regra – 3ª Condição)

Ao avaliar os resultados obtidos, verificou-se que a primeira retornou todas as validações como "Legítimo", enquanto a segunda retornou todas como "Fraude / Erro do sistema 12.3". É importante ressaltar que a primeira *query*

retornou 184 validações, enquanto a segunda retornou 404. Diante destes resultados, podemos inferir que a regra está classificando corretamente as validações para este tipo de tarifa.

8.10. Décima terceira regra

Para avaliar a décima terceiras regras, semelhantemente à regra anterior, foram realizadas alterações nas validações existentes, mais especificamente na coluna "GrupoTitulo". As alterações envolveram a modificação dos valores atribuídos a determinados títulos e suas validações, resultando na atribuição incorreta de um grupo de título em relação ao tipo correspondente. Por exemplo, títulos originalmente pertencentes ao grupo 1 tiveram os seus valores alterados para 2, e aqueles do grupo 2 foram modificados para 1. Essas alterações foram feitas com o objetivo de simular cenários em que os títulos validados estão associados a grupos incorretos. Ao criar esses cenários, de acordo com as condições estabelecidas, espera-se que tais situações sejam identificadas como "Erro do Sistema 13.1" ou "Erro do Sistema 13.2", respetivamente.

O código 31 em MySQL possui duas instruções **UPDATE** têm como objetivo alterar a coluna "GrupoTitulo" da tabela "dataoutput" com base nos valores da coluna "Titulo".

Na primeira, a coluna "GrupoTitulo" é atualizada para o valor 2. Além disso, a condição **WHERE Titulo IN (...)** especifica uma lista de valores específicos de "Titulo" (4817, 4880, 5076, etc.), ou seja, apenas os registos com esses valores serão afetados pela atualização. Por fim, **LIMIT 800** define que, no máximo, 800 registos serão alterados, garantindo que a atualização seja limitada a essa quantidade, mesmo que existam mais registos que atendam aos critérios.

Em relação à segunda, ela muito semelhante à primeira, mas aqui, "GrupoTitulo" é atualizado para o valor 1 e a condição **WHERE Titulo IN (...)** lista um conjunto diferentes de valores de "Titulo" (7, 24, 26, 30, etc.). Relativamente ao limite, foi estabelecido como **LIMIT 1600** que restringe a atualização para, no máximo, 1600 registos, mesmo se houver mais que atendam à condição.

```
update dataoutput set GrupoTitulo = 2 where Titulo in (4817, 4880, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 3744, 3804) limit 800;

update dataoutput set GrupoTitulo = 1 where Titulo in (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 3187, 3193, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857, 4823, 5282, 4829, 4820) limit 1600;
```

Código 31 - Avaliação das classificações (13ª Regra)

A avaliação desta regra seguirá a mesma abordagem aplicada nas regras seis e doze, cujo objetivo já foi previamente detalhado, tornando desnecessária sua repetição neste contexto.

Primeira condição

O código 32 em MySQL contém duas *queries* **SELECT** que têm objetivo consultar a tabela "dataoutput" e identificar registos específicos com base em dois critérios principais: o valor da coluna "GrupoTitulo" e uma lista de valores específicos na coluna "Titulo". Ambas as consultas retornam as colunas "Descricao", "Titulo", "GrupoTitulo" e "Classificacao", mas aplicam filtros diferentes na coluna "GrupoTitulo".

A primeira consulta aplica um filtro na coluna "GrupoTitulo", onde **WHERE GrupoTitulo = 1** seleciona apenas os registos onde o valor de "GrupoTitulo" é 1. Da mesma forma, é aplicado outro filtro na coluna "Filtro", em que **AND Titulo IN (...)** restringe os resultados para que incluam apenas os registos onde o "Titulo" está numa lista específica (4817, 4880, 5076, etc.).

A segunda consulta é semelhante a primeira, a única diferença reside no filtro aplicado na coluna "GrupoTitulo", neste caso, **WHERE GrupoTitulo = 2** filtra apenas os registos onde "GrupoTitulo" é igual a 2.

```
select Descricao, Titulo, GrupoTitulo, Classificacao from dataoutput where GrupoTitulo = 1 and Titulo in (4817, 4880, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 3744, 3804);

select Descricao, Titulo, GrupoTitulo, Classificacao from dataoutput where GrupoTitulo = 2 and Titulo in (4817, 4880, 5076, 4858, 4864, 4870, 4876, 4878, 4885, 4887, 3744, 3804);
```

Código 32 - Avaliação das classificações (13ª Regra – 1ª Condição)

Após a execução das *queries*, constatou-se que a primeira resultou em 754 validações, todas classificadas como "Legítimo", enquanto a segunda retornou 800 validações, sendo todas identificadas como "Erro do Sistema 13.1". Esses resultados indicam que a regra está funcionando corretamente, classificando de forma adequada as validações para esse tipo de tarifa.

Segunda condição

O código 33 em MySQL apresenta uma estrutura muito similar à do código anterior, diferenciando-se apenas pela lista de títulos aplicados. A descrição já apresentada é igualmente válida para este caso, tornando-se desnecessária uma repetição detalhada.

```
select Descricao, Titulo, GrupoTitulo, Classificacao from dataoutput where GrupoTitulo = 2 and Titulo in (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 3187, 3193, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857, 4823, 5282, 4829, 4820);

select Descricao, Titulo, GrupoTitulo, Classificacao from dataoutput where GrupoTitulo = 1 and Titulo in (7, 24, 26, 30, 69, 71, 74, 75, 444, 456, 501, 573, 591, 3187, 3193, 4470, 4486, 4714, 4837, 4839, 4841, 4843, 4849, 4854, 5283, 9003, 9017, 9018, 9019, 9020, 9022, 9023, 9030, 9031, 9032, 9034, 9037, 9044, 9045, 9046, 9051, 9058, 9059, 9060, 9065, 9072, 9073, 9074, 9079, 4780, 4782, 4786, 4857, 4823, 5282, 4829, 4820);
```

Código 33 - Avaliação das classificações (13ª Regra – 2ª Condição)

Durante a análise, verificou-se que a primeira *query* resultou em 1.846 validações, todas categorizadas como "Legítimo", enquanto a segunda apresentou 1.600 validações, classificadas como "Erro do Sistema 12.2". Esses resultados indicam que a regra está operando de forma correta, diferenciando adequadamente as validações para esse tipo de tarifa.

9. Anexo B – Representação gráfica dos resultados da parte 1

Este anexo apresenta os gráficos e visualizações completas que complementam os dados nas tabelas do Capítulo 5.

Cada gráfico corresponde a resultados discutidos nas Partes 1 e 2, incluindo tempos de execução, classificações de validações (legítimas, inválidas, fraudulentas), erros de sistema e outras métricas observadas durante os testes.

Estes elementos visuais permitem uma análise mais detalhada e intuitiva dos dados apresentados.

9.1. Parte 1

O gráfico 1 “Tempos Limpeza dos Dados” apresenta a relação entre a dimensão da amostra (eixo x) e o tempo para a concluir a transformação de limpeza dos dados (eixo y). Observando a distribuição dos pontos e a reta de tendência, nota-se uma ligeira variação nos tempos, mas, alguns estão mais densos próximos ao centro e em torno de 1 minuto e 26 segundos, o que pode indicar que independentemente do tamanho da amostra, o tempo de limpeza dos dados mantém-se relativamente constante, sem grandes desvios que possam ser atribuídos diretamente ao tamanho da amostra. Além disso, o p-valor calculado para esta análise é de 0,762615142, indicando que não há uma correlação estatisticamente significativa entre a dimensão da amostra e o tempo de limpeza dos dados, por outras palavras, a dimensão da amostra não tem um impacto significativo no tempo necessário para limpar os dados. Isso acontece porque o sistema terá de realizar uma análise completa da tabela original, que possui mais de 17 milhões de registos, em cada teste. Assim sendo, a operação de percorrer todos os dados da tabela é o fator determinante do tempo total de processamento, e não a quantidade de dados a serem limpos. Portanto, os tempos de limpeza observados no gráfico são quase iguais, refletindo a natureza invariável do processo de obtenção dos dados da tabela original.

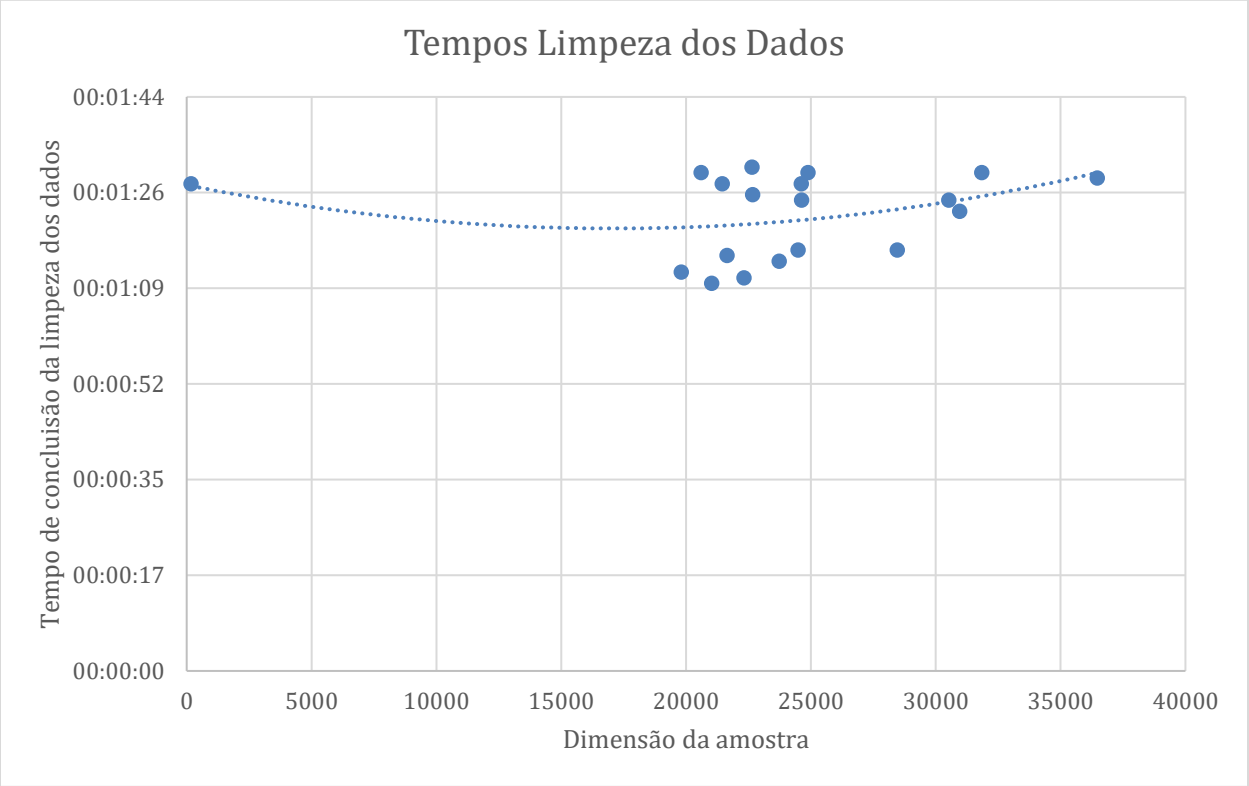


Gráfico 1 – Tempos Limpeza dos Dados

O gráfico 2 “Tempos Manipulação dos Dados” apresenta a relação entre a dimensão da amostra (eixo x) e o tempo para a concluir a transformação de manipulação dos dados (eixo y). Observando a distribuição dos pontos e a linha que começa praticamente no ponto de origem e aumenta de forma não linear à medida que a dimensão aumenta. Isso sugere que o tempo necessário para a manipulação dos dados não aumenta de maneira uniforme à medida que a amostra cresce, mas sim, há um aumento acentuado à medida que a quantidade de dados torna maior. É preciso ter em conta que este aumento acentuado se deve a uma amostra que levou 1 segundo para ser processado. Se o removermos, é provável que a tendência tenha um crescimento muito mais linear. O p-valor calculado para esta análise é de 1,35684E-08 o que é extremamente baixo, o que significa que há uma relação estatisticamente significativa entre a dimensão da amostra e o tempo de manipulação dos dados, ou seja, a probabilidade desse padrão observado ocorrer por acaso é muito baixa. Desta forma, podemos inferir que há uma forte correlação entre estas duas variáveis.

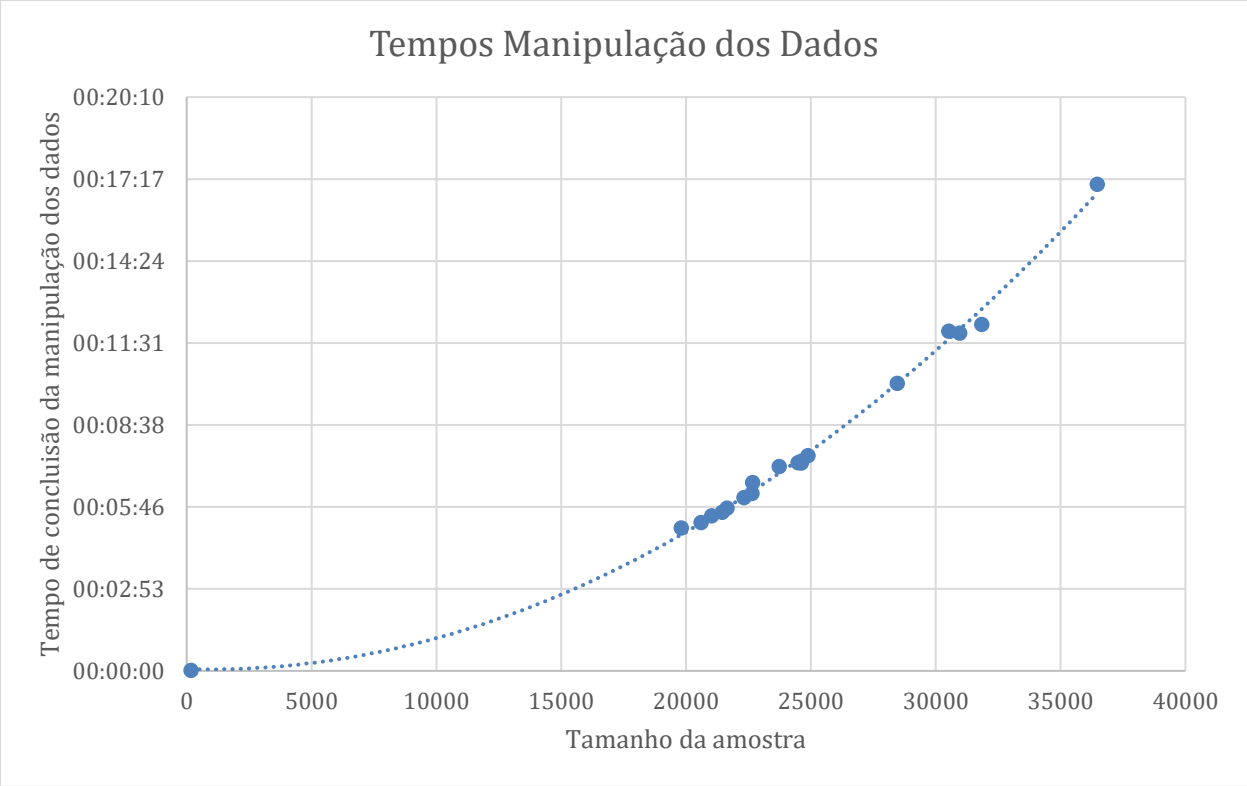


Gráfico 2 - Tempos Manipulação dos Dados

O gráfico 3 “Tempos Classificação dos Dados” exibe a relação entre o tamanho da amostra (eixo x) e o tempo necessário para concluir a transformação de classificação dos dados (eixo y). Semelhantemente ao que foi visto no gráfico anterior, a distribuição dos pontos e da linha sugerem que há um aumento não linear entre o tamanho da amostra e o tempo de classificação. Novamente, o que causa este aumento acentuado é causado pela amostra que possui poucas validações, consequentemente, o tempo de execução foi reduzido. No entanto, se a removermos, possivelmente que a tendência irá ter um crescimento mais linear. O p-valor calculado para esta análise é de 5,75832E-09, trata-se de um valor baixíssimo o que sugere que a relação entre as duas variáveis em causa é fortemente significativa, ou seja, espera-se que à medida que a amostra aumenta o tempo de classificação também vai aumentar.

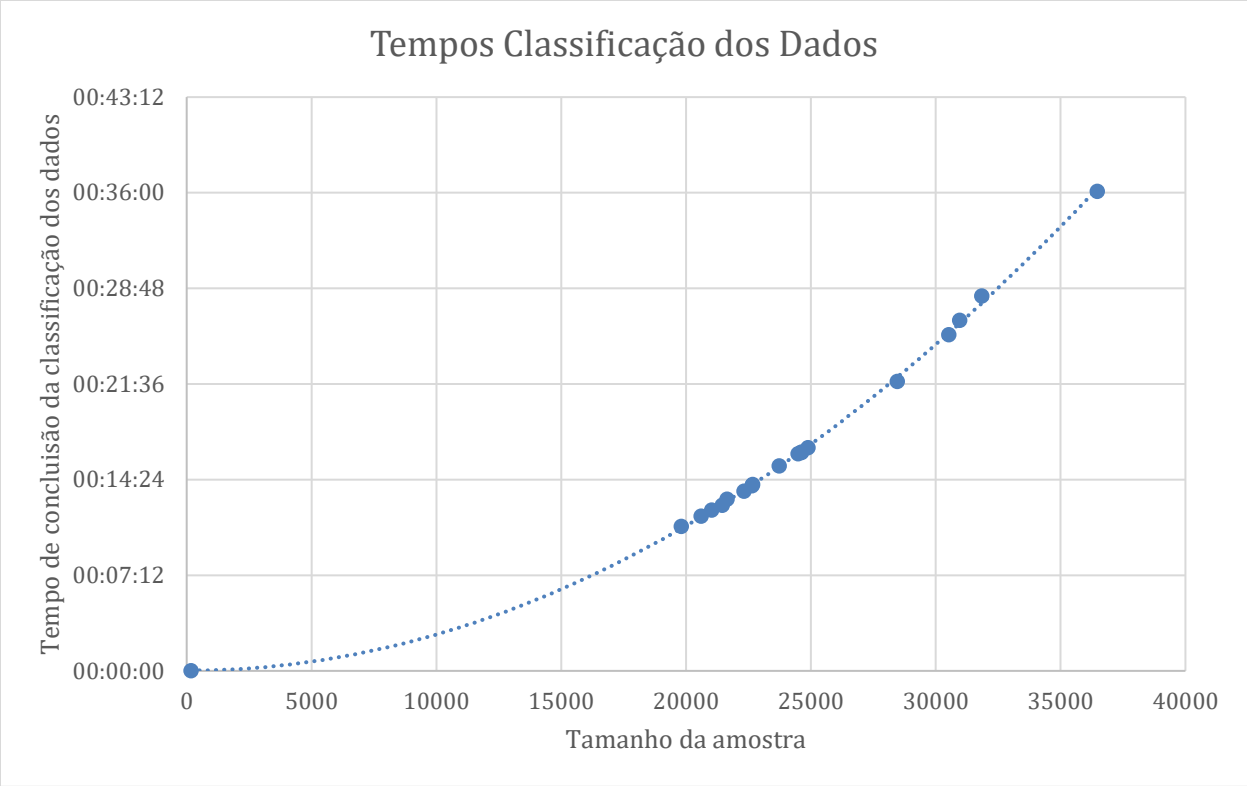


Gráfico 3 - Tempos Classificação dos Dados

O gráfico 4 “Dados em Falta” exibe a relação entre o tamanho da amostra (eixo x) e a quantidade de dados em falta identificados no sistema (eixo y). Se observarmos a distribuição dos pontos, o gráfico revela que, embora o número de casos de erro do sistema apresente uma certa instabilidade – isto é, à medida que o volume de dados aumenta, alguns conjuntos apresentam mais erros do que outros -, a linha de tendência sugere que, à medida que o tamanho da amostra aumenta, a quantidade de erros tende a aumentar. O p-valor calculado para esta análise é de 0,000296487, isso significa que há uma relação forte entre estas duas variáveis, ou seja, podemos concluir que à medida que o tamanho da amostra aumenta, a quantidade de erros detetados espera-se que também aumente. Desta forma, é essencial analisar este gráfico futuramente com mais registos. Se a tendência mostrada se confirmar, será importante resolver esses erros para que a tendência permaneça relativamente contante num valor reduzido.

Neste gráfico apresentado, uma das amostras foi eliminada devido a um valor significativamente mais elevado em relação às demais, o que poderia comprometer a análise dos resultados e não acrescentaria relevância à avaliação. A remoção desta amostra permite uma avaliação mais precisa e representativa do conjunto de dados, evitando distorções que poderiam influenciar negativamente a análise. Esta abordagem também assegura que as tendências observadas são verdadeiras e não são afetadas por *outliers* extremos.

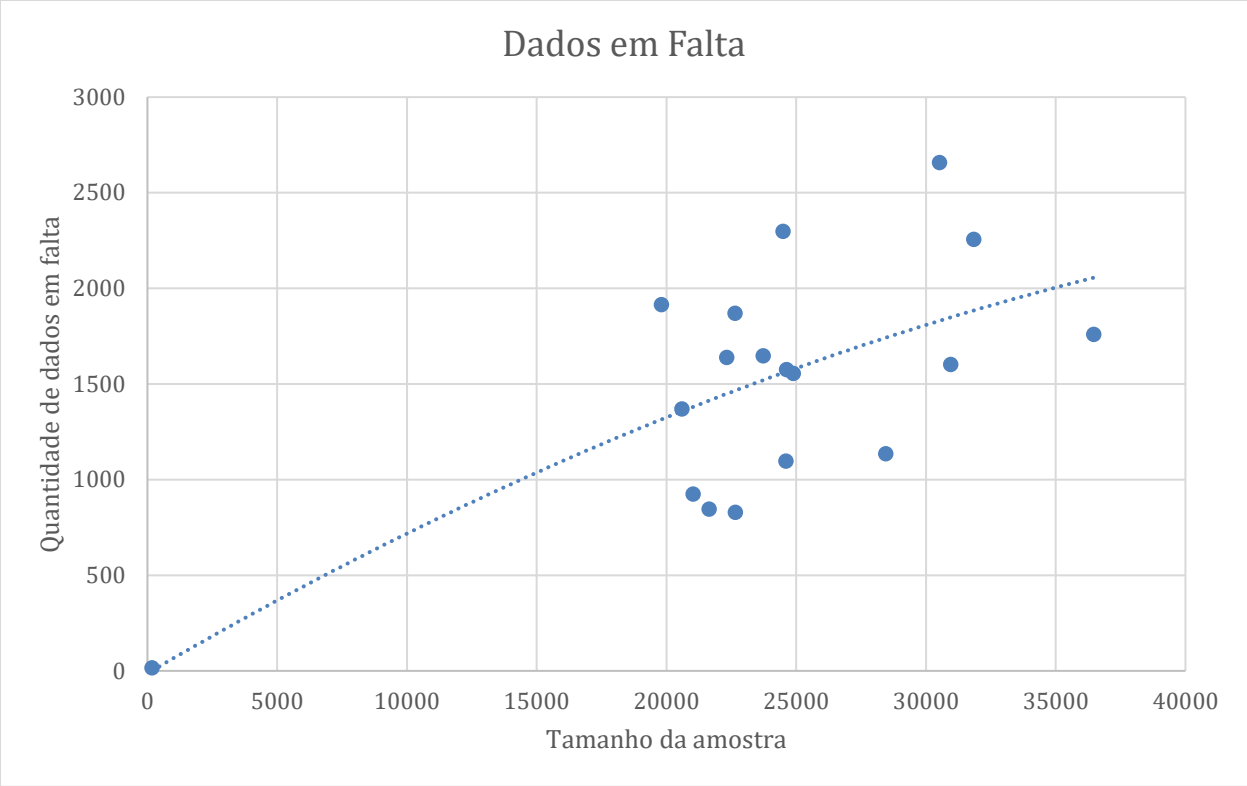


Gráfico 4 - Dados em falta

O gráfico 5 “Erro do Sistema 4” exibe a relação entre o tamanho da amostra (eixo x) e a quantidade de erros identificados no sistema (eixo y). A análise visual indica uma tendência crescente na quantidade de erros conforme aumenta a dimensão da amostra. Além disso, é possível verificar que há uma variabilidade considerável nos erros, mas a maioria dos pontos concentram-se mais em valores mais baixos. Por outro lado, para amostras maiores, há uma tendência clara de aumento no número de erros, com alguns pontos mostrando valores ligeiramente mais altos de erros. O p-valor calculado é de 0,003266058, o que indica que a relação observada entre a dimensão da amostra e a quantidade de erros é estatisticamente significativa, ou seja, sugere que à medida que o tamanho da amostra aumenta, espera-se que a quantidade de erros detetados também aumente. Desta forma, é essencial analisar este gráfico futuramente com mais registos. Se a tendência mostrada se confirmar, será importante resolver esses erros para que a tendência permaneça reduzida e estável.

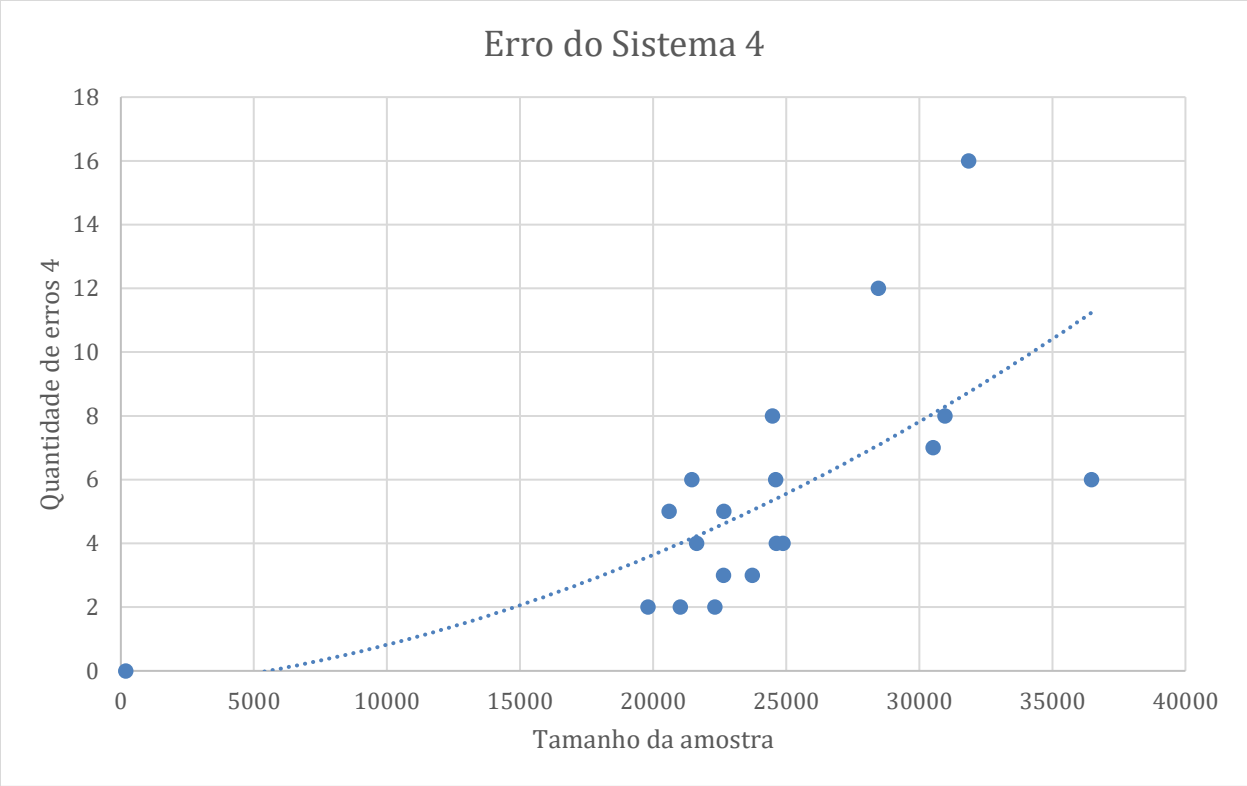


Gráfico 5 - Erro do sistema 4

O gráfico 6 “Fraude 10” mostra a relação entre a dimensão da amostra (eixo x) e a quantidade de fraudes 10 identificadas (eixo y). Observando o gráfico, notamos que, inicialmente, o número de fraudes identificadas aumenta conforme a dimensão da amostra cresce, atingindo um pico em torno dos 25.000 a 30.000 na dimensão da amostra. Após este ponto, a quantidade de fraudes identificadas parece diminuir à medida que a dimensão da amostra continua a aumentar. Além disso, é possível verificar que há uma variabilidade considerável nos erros. O p-valor calculado é de 0,020643307, isso indica que há evidências estatisticamente significativa, ou seja, sugere que a relação entre a dimensão da amostra e a quantidade de fraudes 10 identificadas não é devida ao acaso, mas sim, a dimensão da amostra influencia a quantidade de fraudes 10.

De acordo com o exposto anteriormente, a quantidade de fraudes 10 identificadas, de certa forma, está diretamente relacionada aos “Dados em Falta”. Por outras palavras, quanto mais dados em falta forem identificados, menos fraudes serão detetadas, isso ocorre porque os dados em falta invalidam dados calculados com base neles, que por sua vez, torna-se impossível a classificação como fraudes.

Desta forma, embora o número de fraude 10 é reduzido, tendo em conta o tamanho das amostras, é essencial analisar este gráfico futuramente com mais registos. Se a tendência mostrada se confirmar ou aumentar, será importante resolver esses casos de fraude para que a tendência se torne mais estável num número mais reduzido.

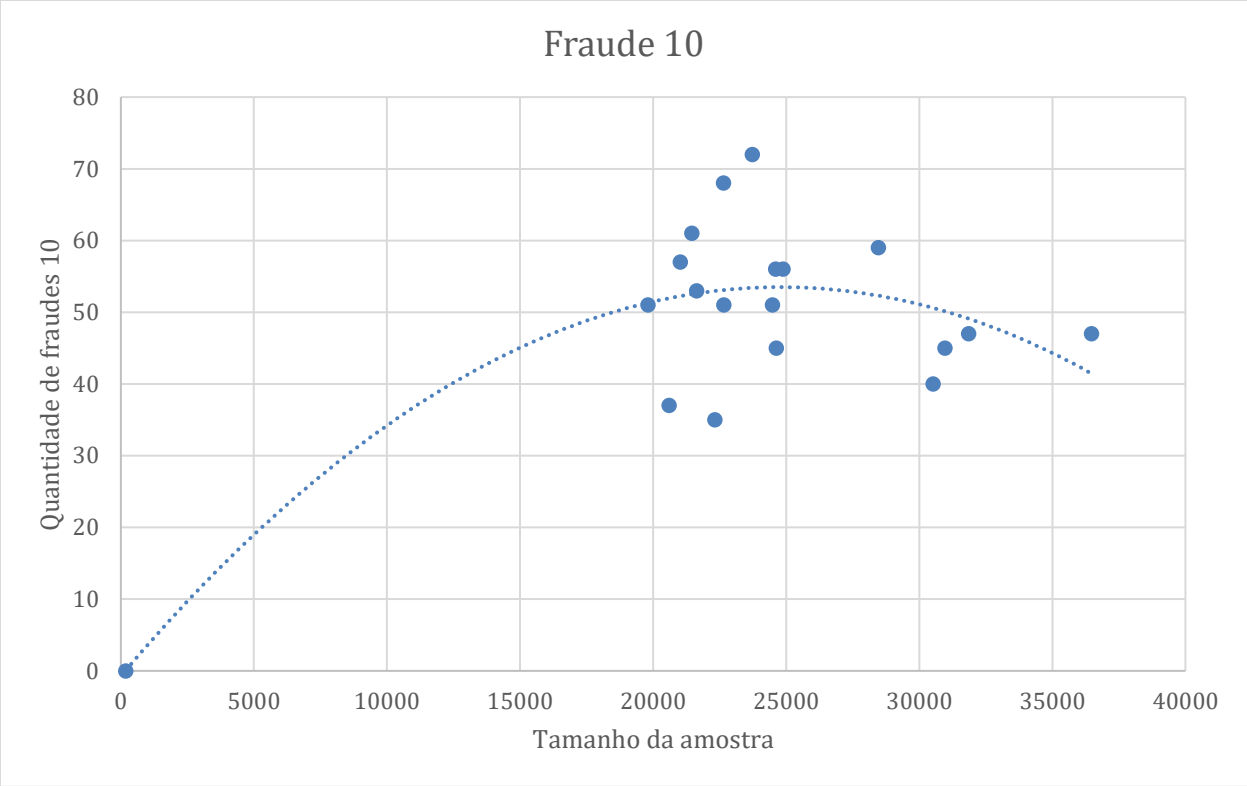


Gráfico 6 - Fraude 10

O gráfico 7 “Inválido” mostra a relação entre a dimensão da amostra (eixo x) e a quantidade de classificações inválidas identificadas (eixo y). Analisando o gráfico, podemos observar uma considerável variabilidade nas classificações inválidas, com a maioria dos pontos concentrando-se em amostras mais baixas. No entanto, para amostras maiores, ao contrário dos gráficos anteriores há uma tendência mais linear crescente e há um aumento claro no número de invalidações, com alguns pontos apresentando valores ligeiramente mais altos. O p-valor calculado é de 0,000731509, o que indica que há uma relação forte entre estas duas variáveis, ou seja, à medida que o tamanho da amostra aumenta, espera-se que a quantidade de classificações inválidas também aumente.

É de relembrar que a classificação inválida ocorre quando não é possível determinar se uma validação é fraudulenta ou legítima. Isso não se deve a um erro do sistema de validações em si, mas sim um valor calculado automaticamente pelo sistema de fraudes com base em dados incorretos provenientes da própria base de dados, que é independente do sistema de fraudes. Dessa forma, a quantidade de classificações inválidas, assim como a ocorrência de fraudes 10, é diretamente influenciada pela quantidade de dados em falta. Isso ocorre, como já foi mencionado, porque os próprios dados em falta geram dados inválidos devido aos cálculos realizados pelo sistema de fraudes, resultando em classificações inválidas. Naturalmente, quanto mais dados em falta ocorrerem, maior será o número de classificações inválidas. Um indicativo disso é a comparação entre o gráfico 7 “Inválido” e o gráfico 4 “Dados em Falta”, que apresentam semelhanças. Sendo assim, tendo em conta o que foi referido, para reduzir o número de classificações inválidas, é necessário resolver/diminuir o número de dados em falta.

Neste gráfico, uma das amostras foi excluída devido a um valor significativamente mais elevado em comparação com as demais. Essa exclusão visa melhorar a análise de dados, garantido uma maior precisão e representatividade no conjunto de dados, mas também, como foi eliminada uma amostra no gráfico 4 “Dados em Falta”, faz sentido também removê-la neste gráfico, dado que estão muito relacionados. Adicionalmente, esta abordagem assegura que as tendências observadas sejam mais genuínas e não sejam afetadas por valores extremos.

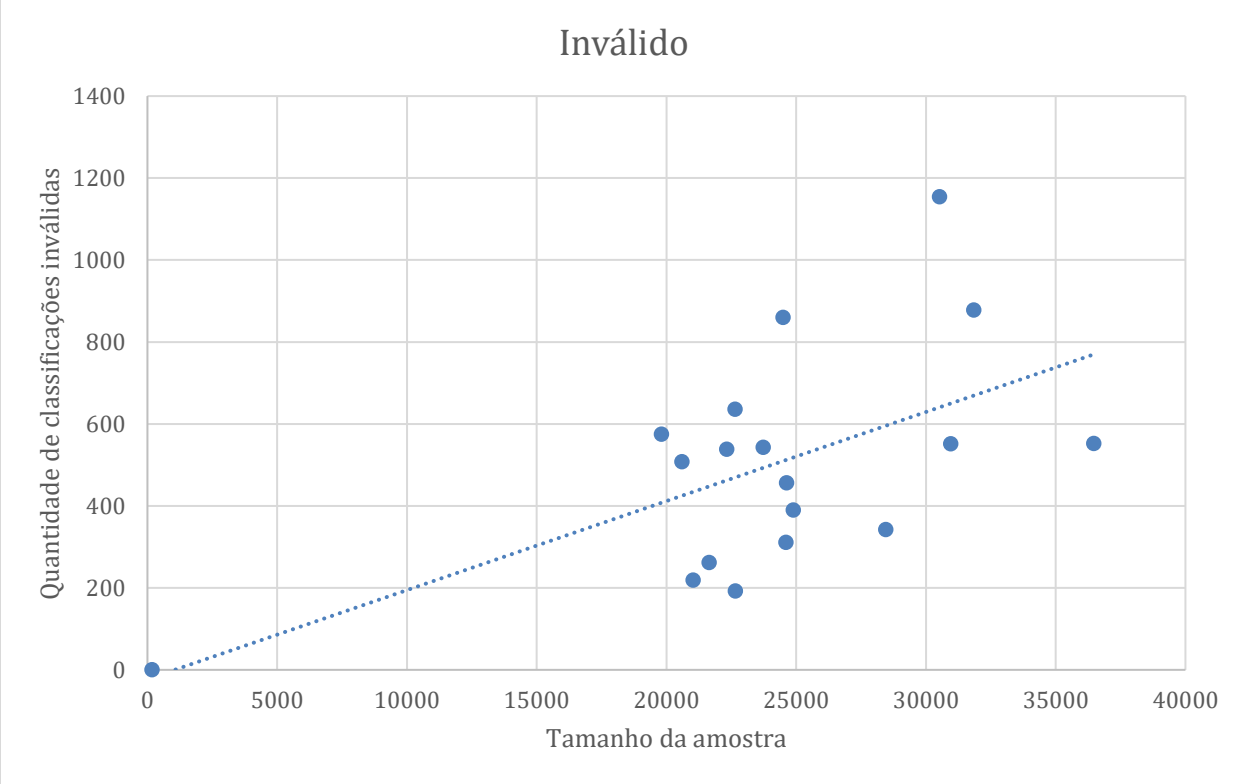


Gráfico 7 - Inválido

O gráfico 8 “Legítimo” exibe uma dispersão de dados onde o eixo x representa a dimensão da amostra, e o eixo y representa a quantidade de casos legítimos identificados em cada amostra. Podemos notar que a linha tendência indica que há uma correlação positiva e crescente quase linear entre as duas variáveis. Além disso, é visível que quase todos os pontos estão próximos da linha x=y, indicando que a grande maioria das validações são legítimas. No entanto, há algumas amostras em que o número de casos legítimos é mais reduzido, geralmente devido às classificações como “Dados em Falta” e casos inválidos, além de outros tipos de erros, suspeitas de fraude, fraudes, etc. O p-valor calculado é extremamente baixo (1,69168E-09), o que significa que a correlação é estatisticamente significativa, por outras palavras, podemos inferir que, à medida que a dimensão da amostra aumenta, é esperado um aumento na quantidade de casos legítimos identificados.

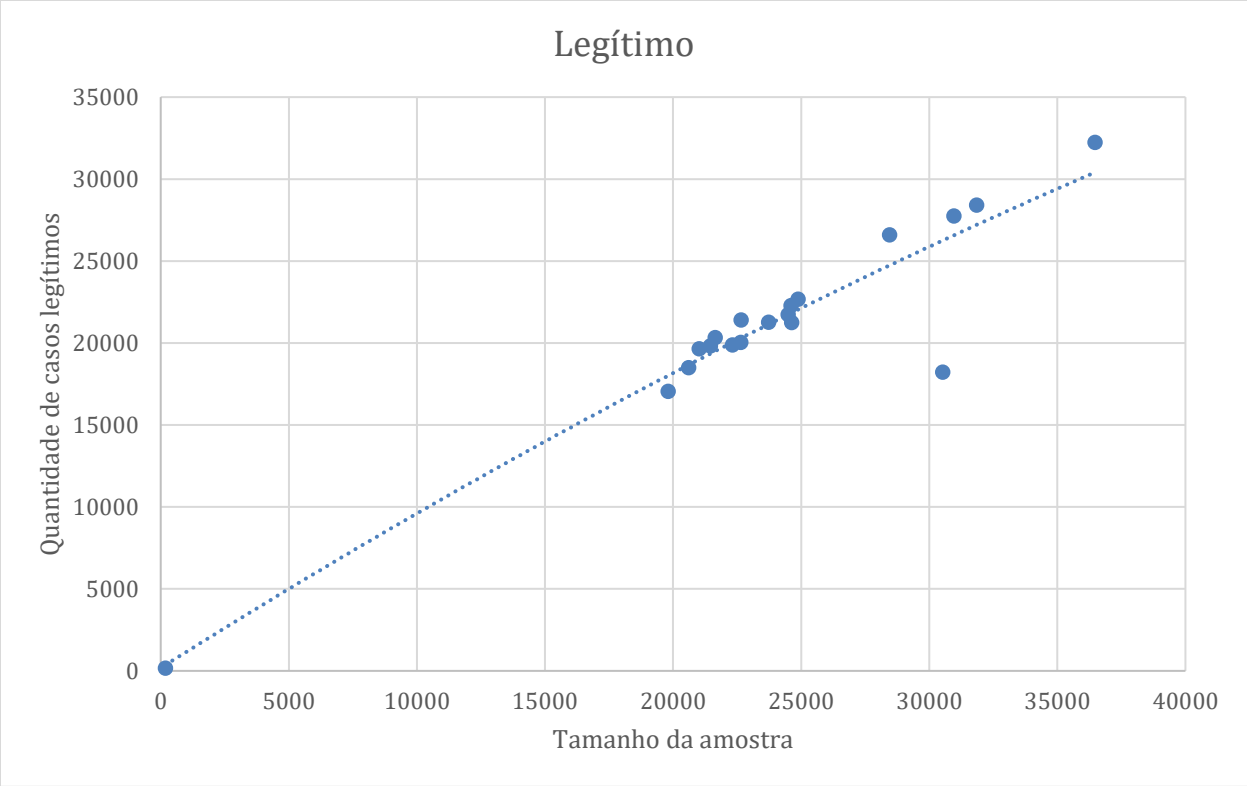


Gráfico 8 - Legítimo

O gráfico 9 “Suspeita de Fraude 9” apresenta uma relação entre o tamanho da amostra (eixo x) e a quantidade de casos de suspeita de fraude 9 identificados (eixo y). A linha de tendência sugere que há um crescimento positivo e relativamente linear, ou seja, à medida que o tamanho da amostra aumenta, mais casos suspeitos de fraude 9 são identificados. Podemos observar ainda uma variabilidade nos casos de suspeita de fraude 9, com a maioria dos pontos concentrando-se em amostras menores. No entanto, à medida que o número de dados aumenta, notamos um claro aumento no número de suspeitas, e alguns pontos apresentando valores ligeiramente mais altos. O p-valor calculado é 1,90733E-07, o que indica uma relação forte entre as duas variáveis, dito de outra forma, à medida que a dimensão da amostra aumenta, é esperado que haja também um aumento no número de suspeitas de fraude 9 identificadas.

Portanto, é crucial examinar este gráfico com mais dados no futuro. Se a tendência indicada se confirmar, será essencial corrigir esses casos de suspeita de fraudes para manter a tendência relativamente estável e com um valor significativamente reduzido.

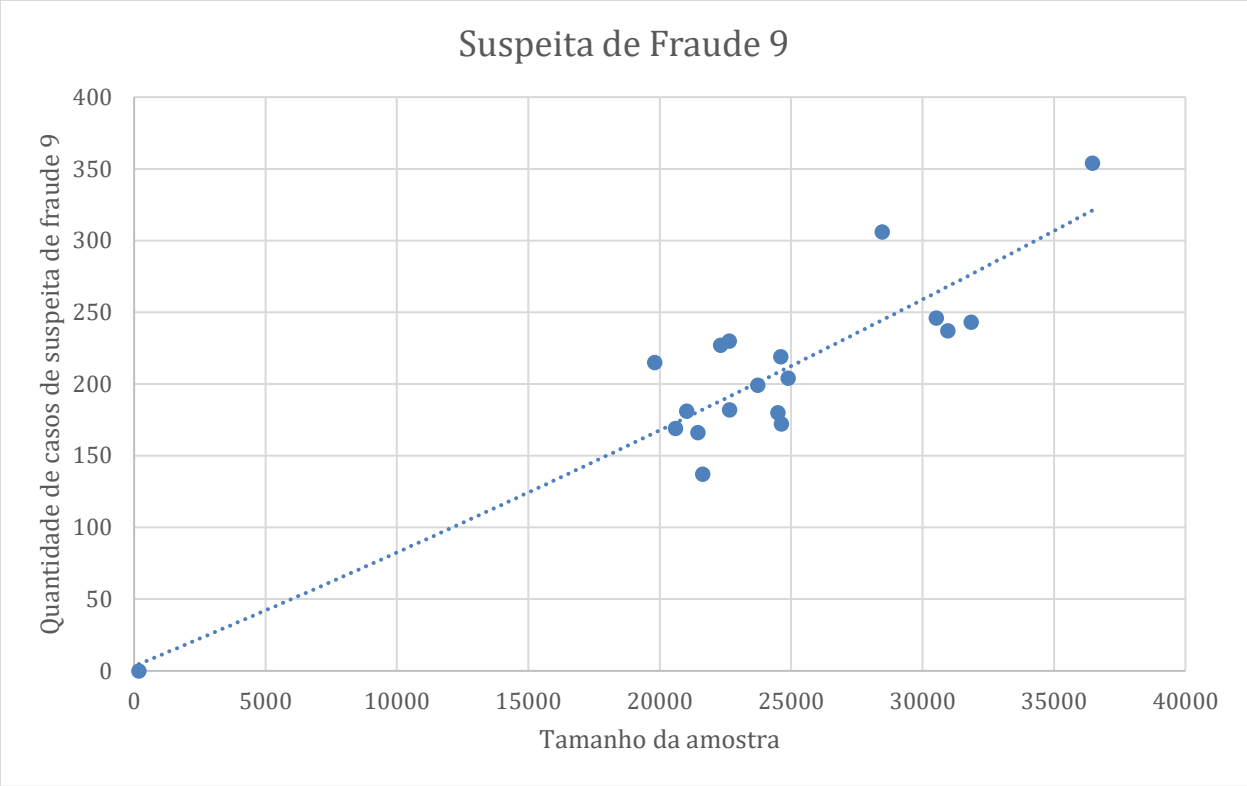


Gráfico 9 - Suspeita de Fraude 9

O gráfico 10 “Suspeita de Fraude 5” representa a relação entre a dimensão da amostra (eixo x) e a quantidade de casos suspeitos de fraude 5 (eixo y). Podemos observar que a maioria das amostras tem 0 casos suspeitos, no entanto, há algumas com um ou dois casos, e só uma apresenta 3. O p-valor calculado é 0,762631135, indicando que não há uma correlação estatisticamente significativa entre as duas variáveis aqui em causa, por outras palavras, a dimensão da amostra não tem um impacto significativo na quantidade de casos suspeitos 5 identificados. Isso ocorre porque suspeitas deste tipo são raras e não foram identificados casos suspeitos de fraude 5 na maioria das amostras.

No futuro, com mais resultados, a distribuição dos casos suspeitos pode comportar-se de maneira diferente, afetando a tendência e o p-valor. Além disso, é importante examinar se a tendência indicada se confirma, em caso negativo, será essencial corrigir esses casos de suspeita de fraude para manter a tendência relativamente estável e com valores bastantes reduzidos, como ilustrado neste gráfico.

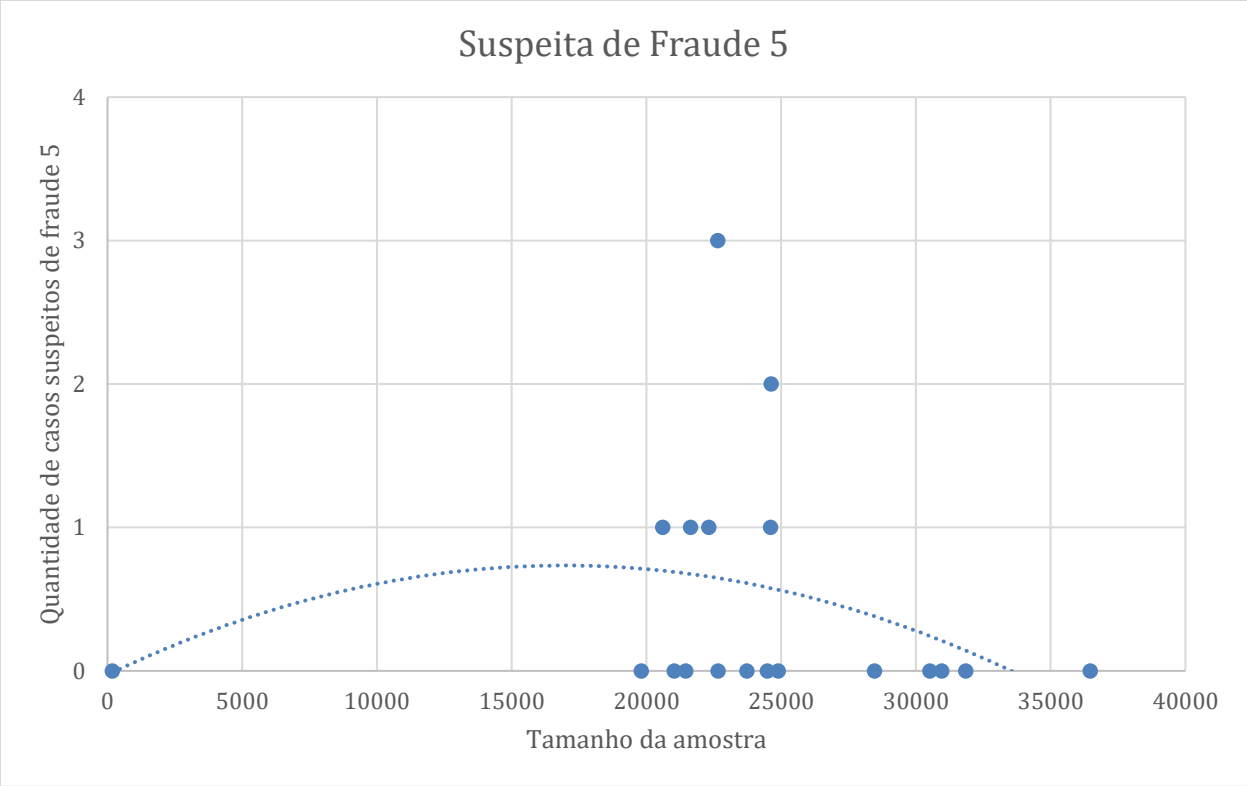


Gráfico 10 - Suspeita de Fraude 5

O gráfico 11 “Fraude / Erro do Sistema 6.1” ilustra a relação entre a dimensão da amostra (eixo x) e a quantidade de casos de Fraude / Erro do Sistema 6.1 (eixo y), além disso, podemos observar que todas as amostras não possuem casos. O p-valor obtido é de 0,660835435, indicando que não há correlação estatisticamente significativa entre as duas variáveis. Por outras palavras, a dimensão da amostra não influência de forma significativa a quantidade de casos de fraude / erros do sistema 6.1 identificados. Isso ocorre porque a maioria das amostras não apresentou qualquer caso aqui em causa.

Como já foi mencionado anteriormente, A classificação “Fraude / Erro do Sistema 6.1” ocorre quando um bilhete pré-comprado de multiviagens apresenta uma quantidade de minutos de viagem superior ao permitido. Por exemplo, um bilhete diário pré-comprado que possui 10.000 minutos no momento da validação, o que não é permitido. Isso é um caso raro, possivelmente resultante de um *bug* durante a criação do bilhete ou de uma manipulação intencional.

Diferentemente dos gráficos anteriores, não é recomendável eliminar a amostra que apresenta um número mais alto de casos identificados. Isso ocorre porque, ao retirar essa amostra, o gráfico perde a sua relevância, uma vez que as restantes amostras não apresentam casos de “Fraude / Erro do Sistema 6.1”.

No futuro, com mais dados, a distribuição pode sofrer alterações, o que pode afetar a tendência e o p-valor. Além disso, é importante analisar se a tendência observada se mantém. Caso contrário, será crucial corrigir estes casos para manter a tendência relativamente estável e com valores bastantes baixos, conforme mostrado neste gráfico.

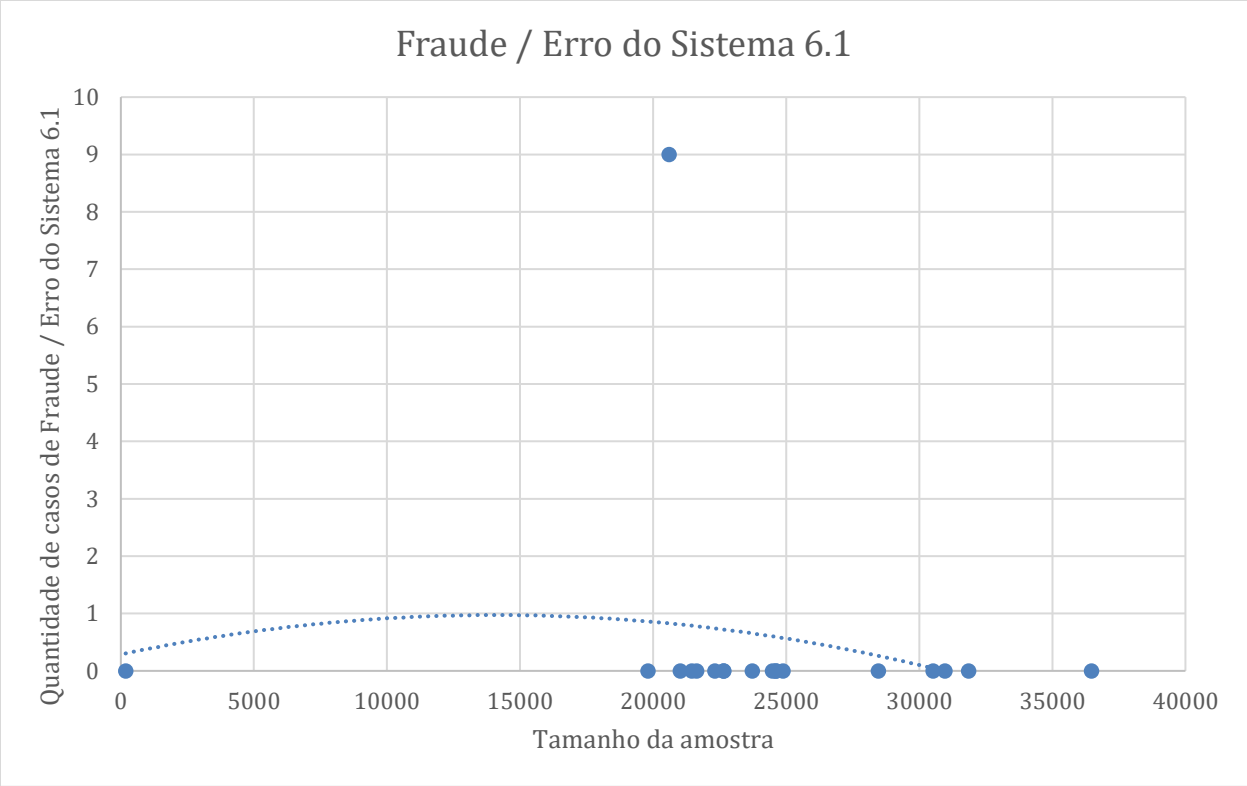


Gráfico 11 - Fraude / Erro do Sistema 6.1

9.2. Parte 2

O gráfico 12 “Tempos Limpeza dos Dados” ilustra a relação entre o tamanho da amostra (eixo x) e o tempo gasto para a limpeza dos dados (eixo y). Ao analisar os pontos e a reta de tendência, observa-se uma pequena variação nos tempos, indicando que o tempo de limpeza dos dados permanece quase constante, independentemente do tamanho da amostra. O p-valor obtido nesta análise é de 0,755618297, sugerindo que não há correlação estatisticamente significativa entre o tamanho da amostra e o tempo de limpeza dos dados. Em outras palavras, a dimensão da amostra não influencia de forma significativa o tempo de conclusão da limpeza. Isso se deve ao facto de que, em cada limpeza, o sistema precisa analisar completamente a tabela original, que contém mais de 17 milhões de registos. Portanto, o fator determinante do tempo total de processamento é a necessidade de percorrer todos os dados da tabela, e não a quantidade de dados a serem limpos. Consequentemente, os tempos de limpeza observados no gráfico são praticamente iguais, refletindo a invariabilidade do processo de obtenção dos dados da tabela original.

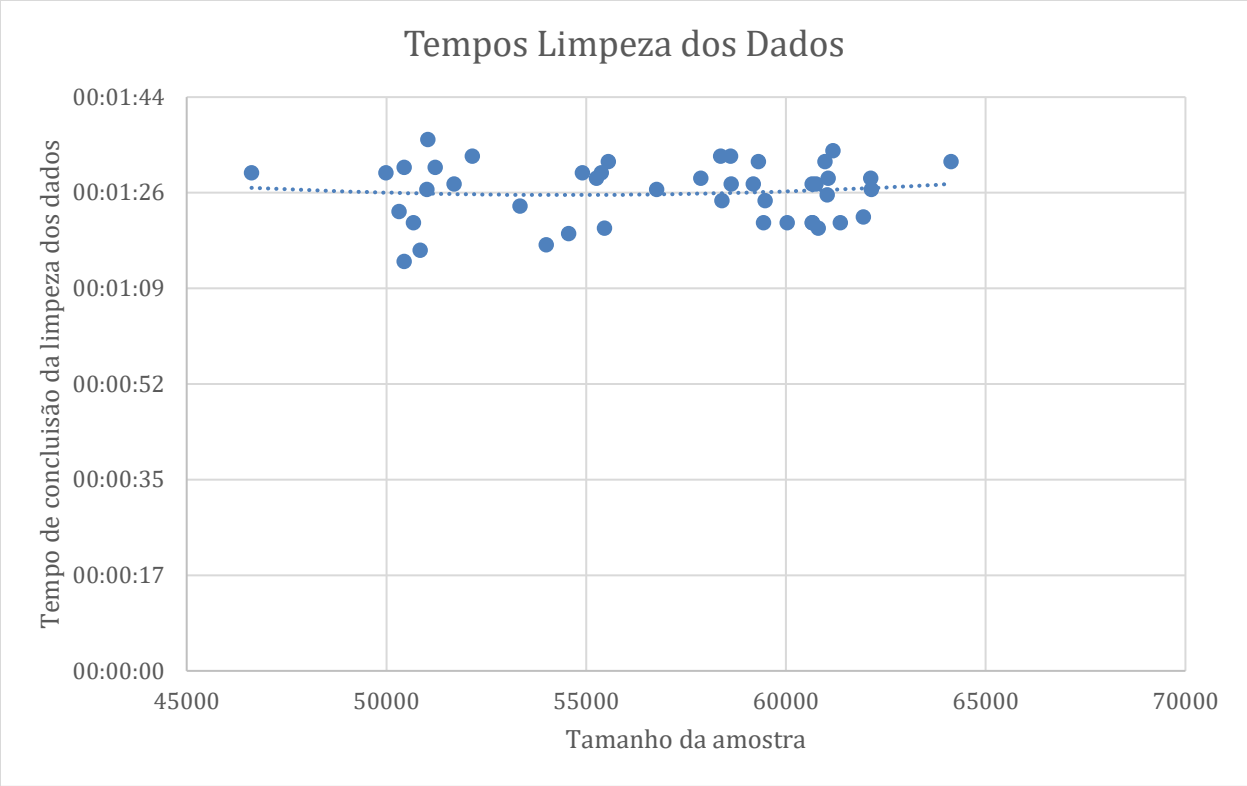


Gráfico 12 - Tempos Limpeza dos Dados

O gráfico 13 intitulado “Tempos Manipulação dos Dados” demonstra a relação entre o tamanho da amostra (eixo x) e o tempo necessário para concluir a transformação de manipulação dos dados (eixo y). Ao analisar a distribuição dos dados e a reta de tendência, observa-se um aumento quase linear conforme a dimensão da amostra cresce. Isso indica que o tempo necessário para manipulação dos dados aumenta de maneira quase uniforme com o aumento do tamanho da amostra. O p-valor calculado para esta análise é extremamente baixo, 1,31831E-43, indicando uma relação estatisticamente significativa entre o tamanho da amostra e o tempo de manipulação dos dados, ou seja, a probabilidade de esse padrão observado ser devido ao acaso é muito pequena. Assim, podemos concluir que existe uma forte correlação entre estas duas variáveis.

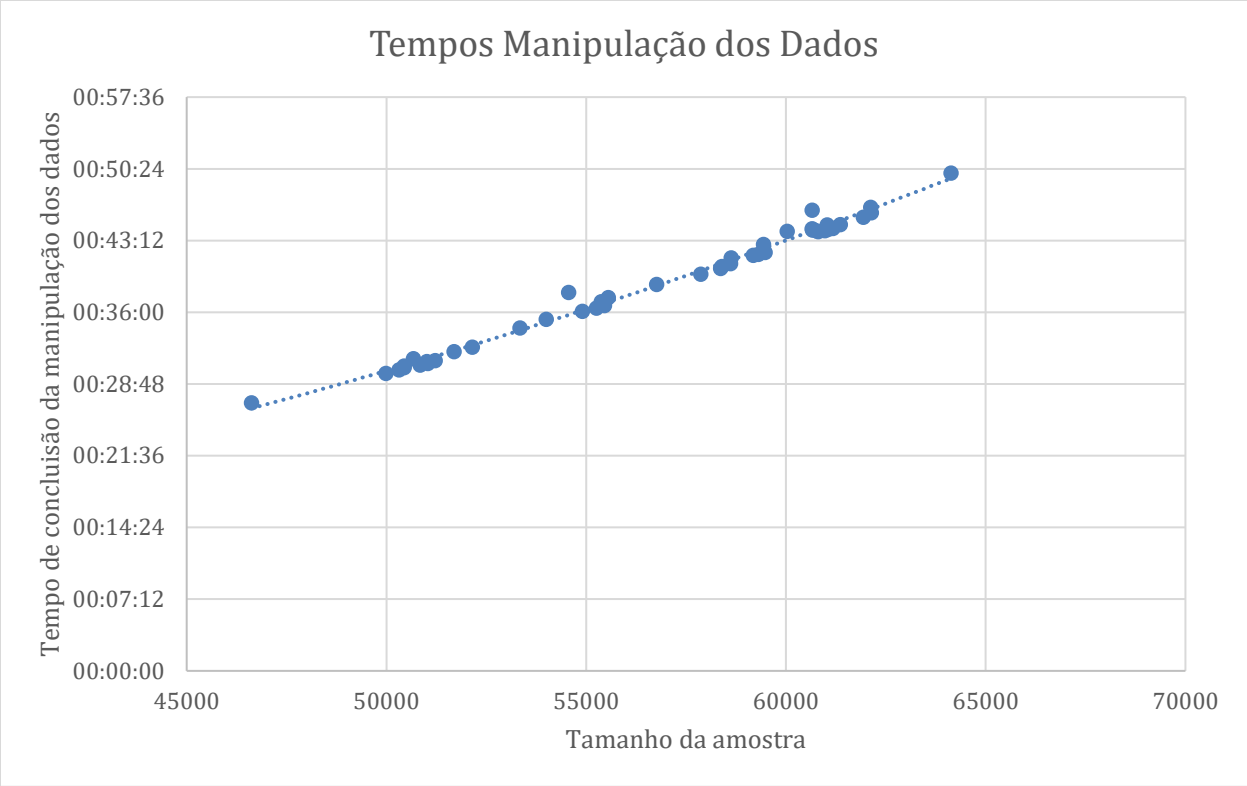


Gráfico 13 - Tempos Manipulação dos Dados

O gráfico 14 “Tempos Classificação dos Dados” mostra a relação entre o tamanho da amostra (eixo x) e o tempo necessário para concluir a transformação de classificação dos dados (eixo y). A disposição dos pontos e da linha de tendência indicam que, conforme o tamanho da amostra aumenta, o tempo de classificação também aumenta de maneira quase linear, similar ao observado no gráfico anterior. O p-valor calculado para esta análise é extramente baixo, 2,41099E-49, o que sugere uma relação fortemente significativa entre as duas variáveis. Isso significa que, à medida que a amostra cresce, espera-se que o tempo de classificação também aumente.

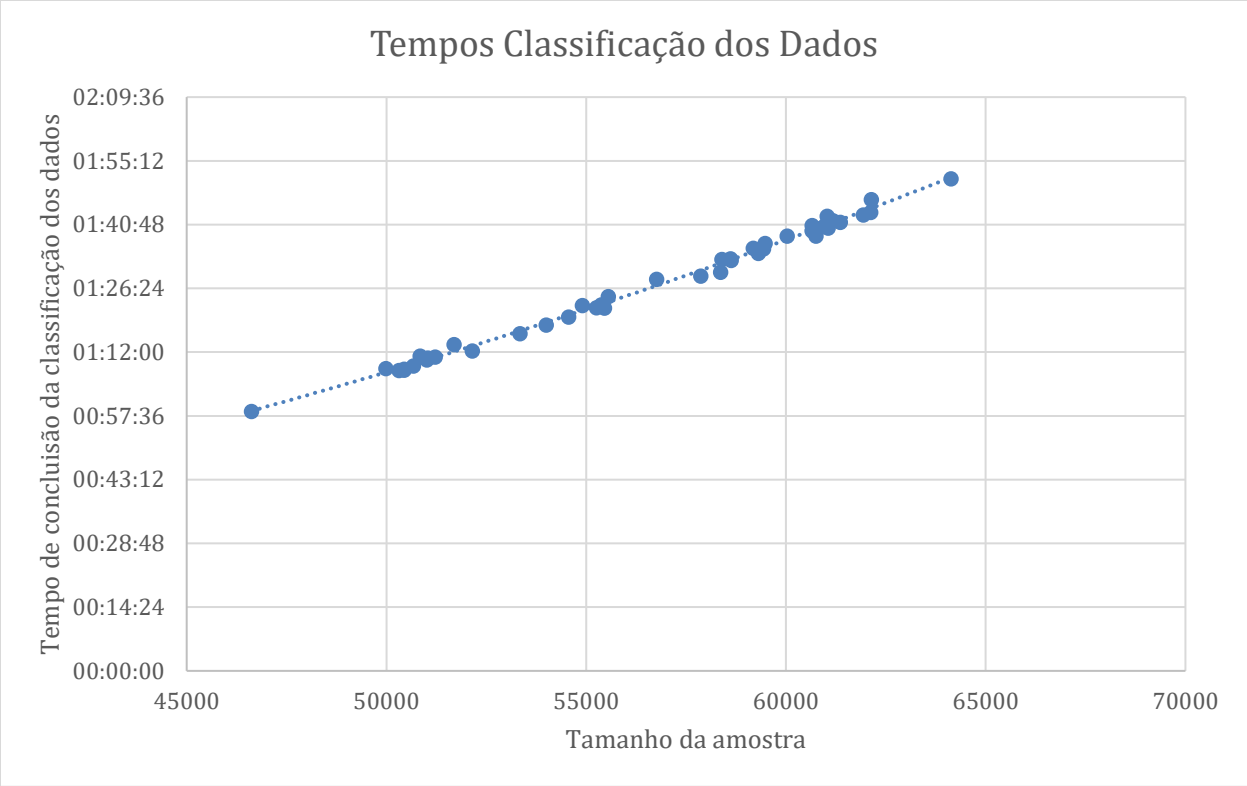


Gráfico 14 - Tempos Classificação dos Dados

O gráfico 15 “Dados em Falta” representa a relação entre o tamanho da amostra (eixo x) e a quantidade de dados em falta identificados no sistema (eixo y). A distribuição de pontos revela alguma instabilidade e disparidade à medida que o volume de dados aumenta, onde alguns conjuntos apresentam mais erros do que outros. A linha tendência sugere que, à medida que o tamanho da amostra aumenta, a quantidade de erros tende a crescer. O p-valor calculado para esta análise é de 0,000508076, indicando uma forte relação entre as variáveis. Em resumo, à medida que o tamanho da amostra aumenta, espera-se que a quantidade de erros detetados também aumente. É importante continuar a analisar este gráfico com mais dados no futuro. Se a tendência persistir, será crucial corrigir esses erros para manter a estabilidade em valores bastantes reduzidos.

No gráfico apresentado, uma amostra foi excluída devido a um valor significativamente mais alto em relação às outras e que não tem grande relevância para a análise. Essa exclusão garante uma avaliação mais precisa e representativa do conjunto de dados, evitando distorções causadas por *outliers* extremos.

É importante destacar que a razão para a adoção deste tipo de classificação, ou seja, o contexto em que ela surge, já foi abordada anteriormente.

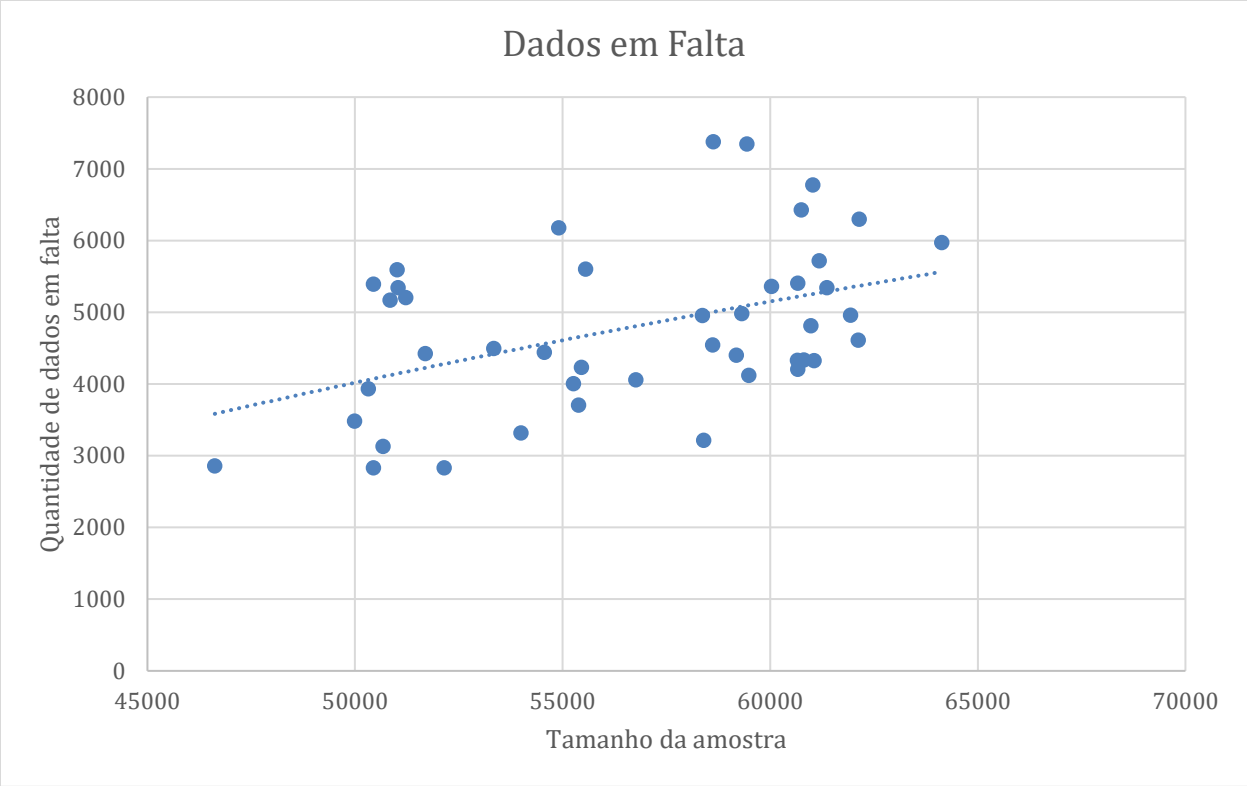


Gráfico 15 - Dados em Falta

O gráfico 16 “Erro do Sistema 4” apresenta a relação entre o tamanho da amostra (eixo x) e a quantidade de erros identificados no sistema (eixo y). Inicialmente, a linha de tendência, mostra que há uma diminuição do número de erros identificados conforme o tamanho da amostra aumenta, atingindo um mínimo em torno de 55.000 a 60.000 na dimensão da amostra. Após esse ponto, a quantidade de erros tende a aumentar conforme a amostra continua a crescer. Além disso, há uma variabilidade considerável nos erros. Para amostras maiores, observa-se uma tendência de aumento no número de erros, com alguns pontos mostrando valores ligeiramente mais altos. No entanto, o p-valor calculado é de 0,59412249, indicando que a relação observada entre o tamanho da amostra e a quantidade de erros não é estatisticamente significativa, ou seja, sugere que, à medida que o tamanho da amostra aumenta, não se espera grandes variações na quantidade de erros detetados.

Portanto, é crucial realizar uma análise mais detalhada deste gráfico com um maior número de amostras no futuro. Caso a tendência para amostras maiores aumente, será fundamental corrigir esses erros para manter a estabilidade em valores consistentemente baixos.

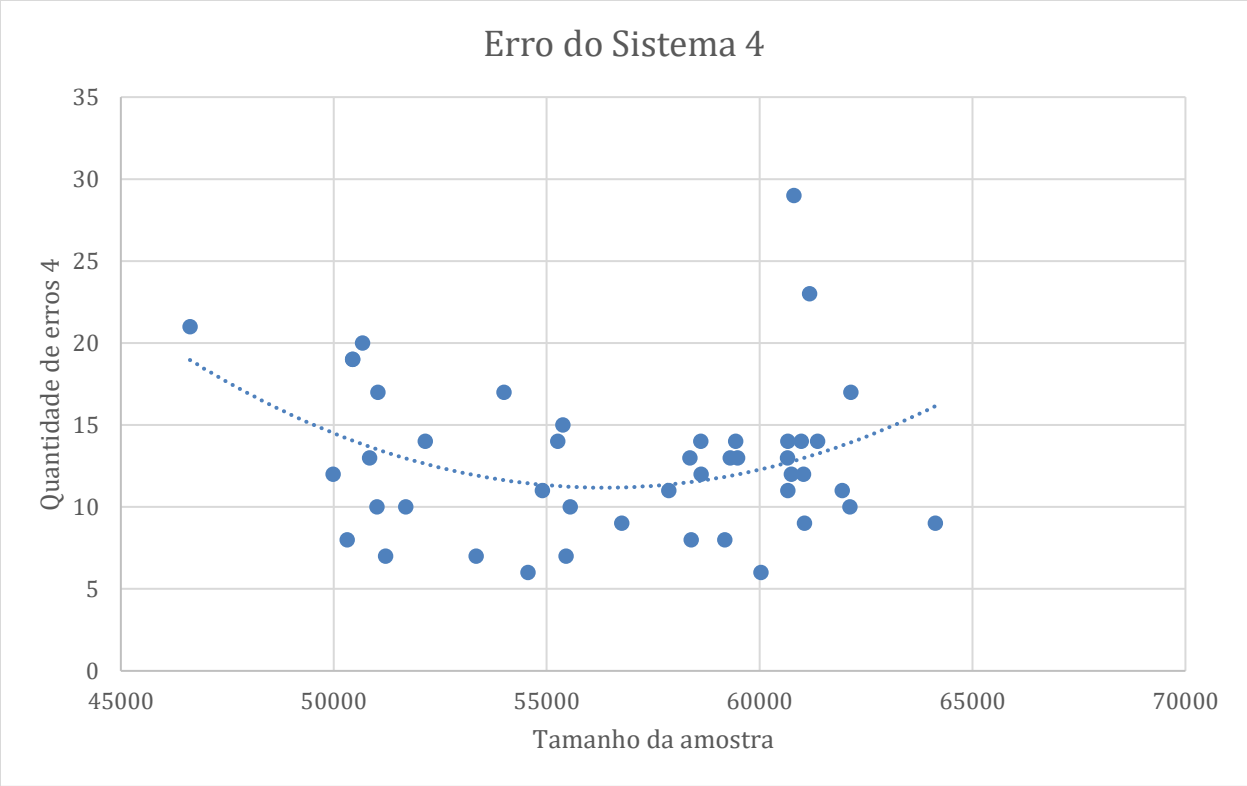


Gráfico 16 - Erro do Sistema 4

O gráfico 17 “Fraude 10” mostra a relação entre o tamanho da amostra (eixo x) e a quantidade de fraudes 10 identificadas (eixo y). Ao observar o gráfico, percebemos que inicialmente o número de casos identificados aumenta conforme o tamanho da amostra cresce, estabilizando em torno das 60.000 validações classificadas no total. Além disso, é possível presenciar uma considerável variabilidade nos resultados. O p-valor calculado é de 0,060666622, o que indica que não há evidência estatisticamente significativa de que o tamanho da amostra influencia a quantidade de fraude 10 identificadas. Contudo, a tendência constante apresentada pela linha não é completamente confiável, pois com mais dados no futuro, especialmente na faixa entre 60.000 e 65.000 dados, é provável que ela se altere para um crescimento positivo.

Convém ainda mencionar que a relação desta classificação com a “Dados em Falta” já foi discutida anteriormente, sendo desnecessário expor novamente detalhadamente a associação entre elas.

Portanto, mesmo com um número relativamente reduzido de fraudes 10, tendo em conta o tamanho da amostra, é crucial examinar este gráfico com mais registos no futuro. Caso a tendência observada se confirme ou cresça, será fundamental abordar esses casos de fraude para manter a estabilidade em valores baixos.

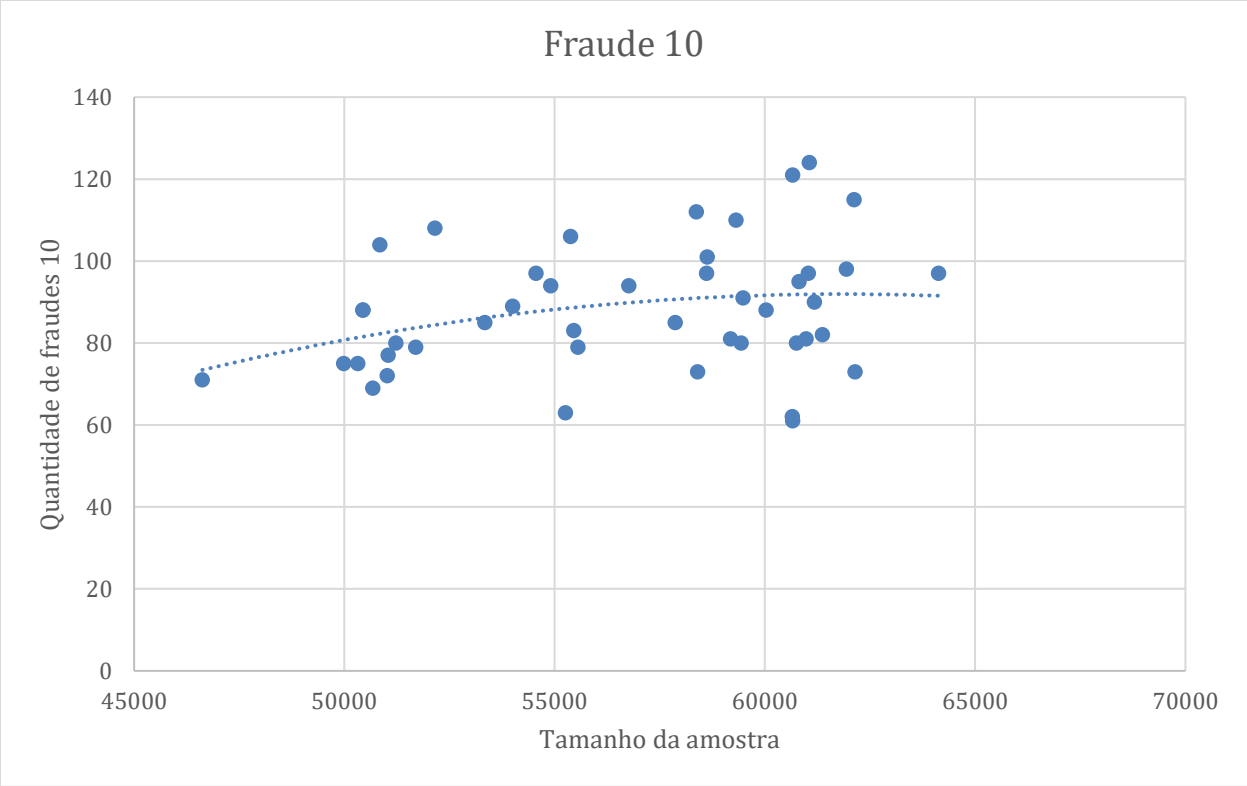


Gráfico 17 - Fraude 10

O gráfico 18 “Inválido” ilustra a relação entre o tamanho da amostra (eixo x) e a quantidade de classificações inválidas identificadas (eixo y). Vale relembrar que uma classificação é considerada inválida quando não é possível determinar se uma validação é legítima ou fraudulenta. Ao analisar o gráfico, observamos uma variabilidade considerável nas classificações inválidas, com a maioria dos pontos estarem ligeiramente mais concentrados em amostras maiores. No entanto, percebemos que inicialmente o número de invalidações identificadas aumenta conforme o tamanho da amostra cresce, diminuindo em torno dos 60.000. O p-valor calculado é de 0,03833853, indicando que há evidência estatisticamente significativa de que o tamanho da amostra influencia a quantidade de classificações inválidas, ou seja, à medida que o tamanho da amostra aumenta, espera-se que haja um aumento no número de invalidações.

A relação entre a quantidade de classificações inválidas e a ocorrência de “Dados em Falta” já foi abordada anteriormente, tornando desnecessária a repetição desse ponto.

Adicionalmente, foi excluído uma amostra com um valor muito mais elevado, não só para melhorar a análise de dados, garantindo uma maior precisão e uma melhor representação dos conjuntos dos dados, mas também, como foi eliminada uma amostra no gráfico “Dados em Falta”, faz sentido também removê-la neste gráfico, dado que estão muito relacionados. Desta forma, esta abordagem assegura que as tendências observadas não sejam afetadas por valores extremos.

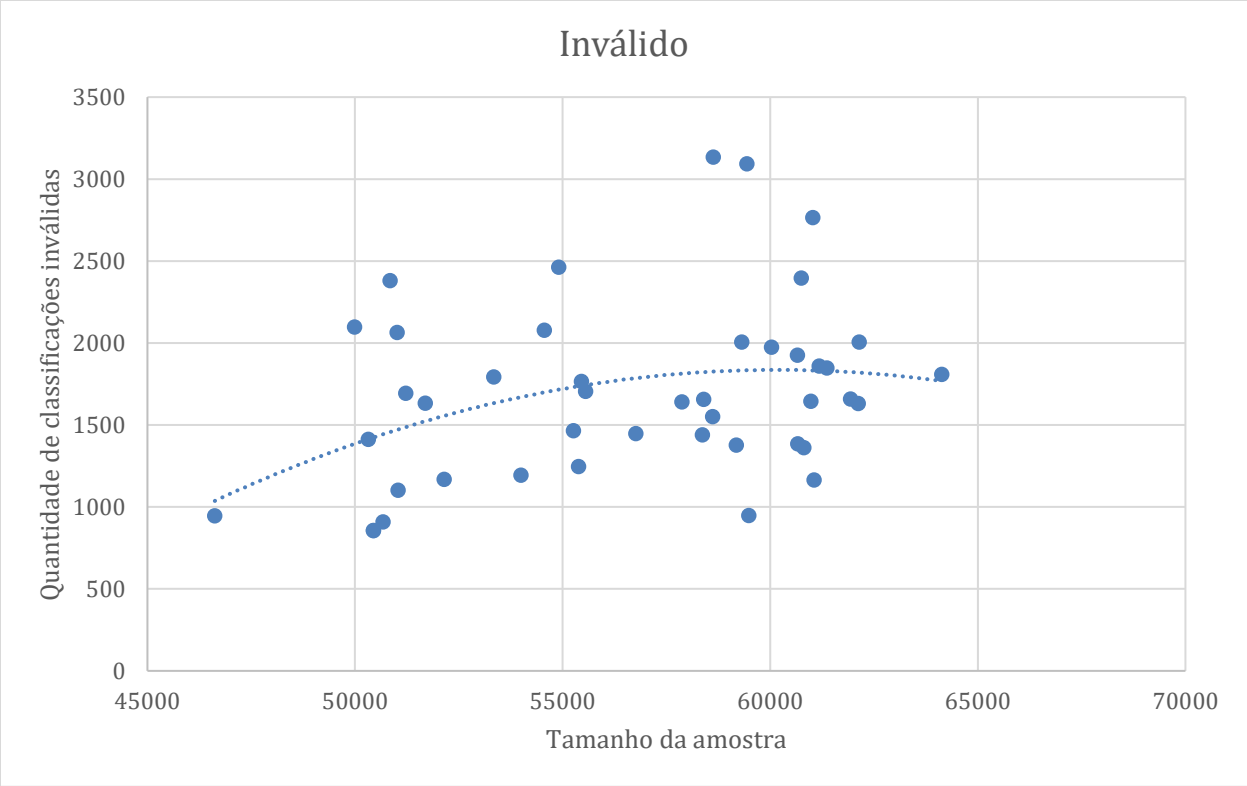


Gráfico 18 - Inválido

O gráfico 19 “Legítimo” mostra uma dispersão de dados, onde o eixo x representa o tamanho da amostra e o eixo y representa a quantidade de casos legítimos identificados em cada amostra. Observa-se que a linha de tendência sugere um crescimento positiva e quase linear entre estas variáveis. Mas, em algumas amostras, o número de casos legítimos é menor, geralmente devido a “Dados em Falta”, casos inválidos, e outros tipos de problemas como suspeitas de fraude, como já é sabido. O p-valor calculado é muito baixo (3,64971E-11), indicando uma correlação estatisticamente significativa. Dessa forma, podemos concluir que conforme o tamanho da amostra aumenta, espera-se um aumento na quantidade de casos legítimos identificados.

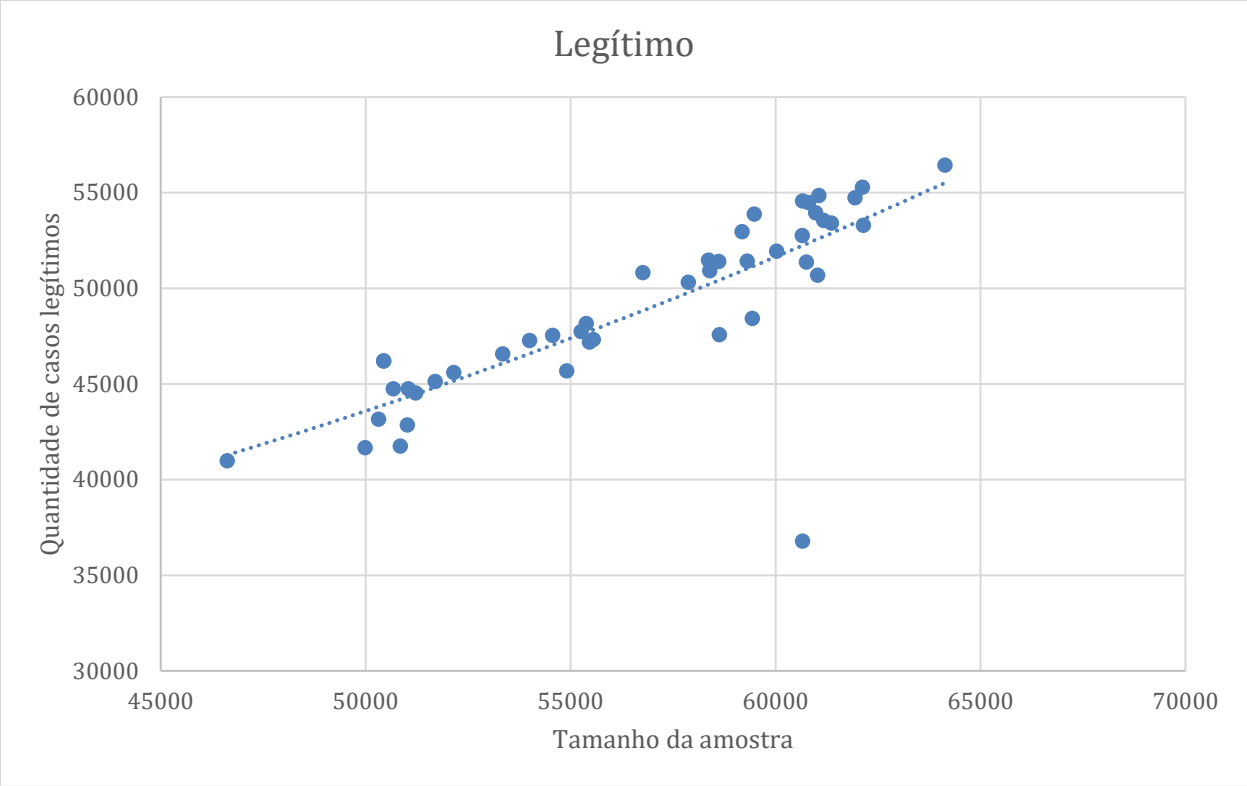


Gráfico 19 - Legítimo

O gráfico 20 “Suspeita de Fraude 9” ilustra a relação entre o tamanho da amostra (eixo x) e a quantidade de casos identificados como suspeita de fraude 9 (eixo y). A linha de tendência e distribuição dos pontos mostram um crescimento positivo relativamente linear e uma variabilidade nos dados, indicando que à medida que o tamanho da amostra aumenta, mais casos de suspeita de fraude 9 são detetados. Adicionalmente, é possível notar que à medida que o número de dados aumenta, o número de casos de suspeita aumenta ligeiramente. Contudo, o p-valor calculado é 0,105040092, sugerindo que não há uma relação forte entre as duas variáveis, por outras palavras, à medida que o tamanho da amostra aumenta, não se espera uma variação substancial na quantidade de suspeitas de fraude 9 identificadas. No entanto, é fundamental realizar uma análise mais abrangente deste gráfico com dados adicionais no futuro, pois o p-valor e a linha de tendência podem sofrer alterações.

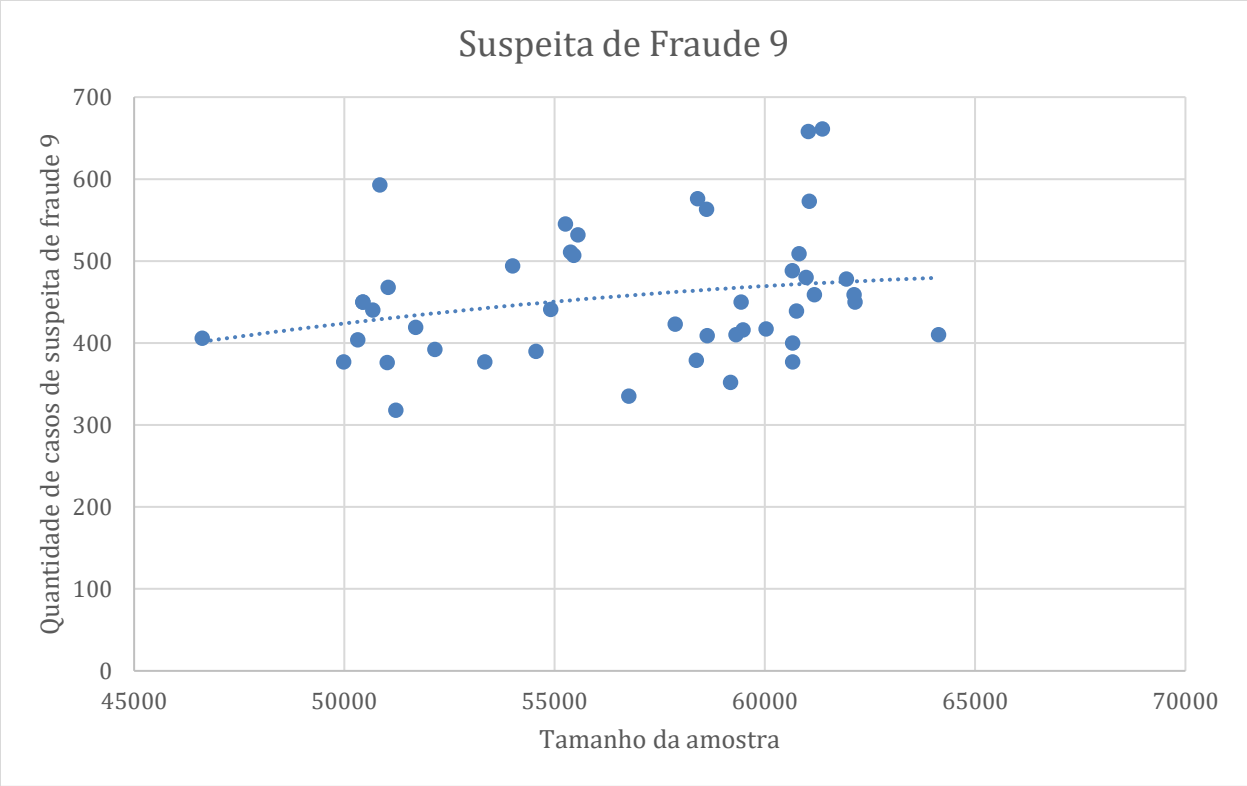


Gráfico 20 - Suspeita de Fraude 9

O gráfico 21 “Suspeita de Fraude 5” ilustra a relação entre o tamanho da amostra (eixo x) e a quantidade de casos suspeitos de fraude 5 (eixo y). Observa-se que a maioria das amostras apresenta 0 casos suspeitos, embora algumas contenham 1 caso. O p-valor calculado é 0,990563758, indicando que não há uma correlação estatisticamente significativa entre as duas variáveis. Por outras palavras, o tamanho da amostra não tem um impacto relevante na quantidade de casos suspeitos de fraude 5 identificados. Isso se deve à raridade desse tipo de suspeita, com quase nenhuma ocorrência em praticamente todas as amostras.

No futuro, com mais amostras, a distribuição dos casos suspeitos pode comportar-se de maneira diferentes, afetando a tendência e o p-valor. Além disso, será importante verificar se a tendência observada se mantém. Caso contrário, será necessário corrigir esses casos de suspeita de fraude para manter a estabilidade e garantir valores baixos, conforme ilustrado neste gráfico.

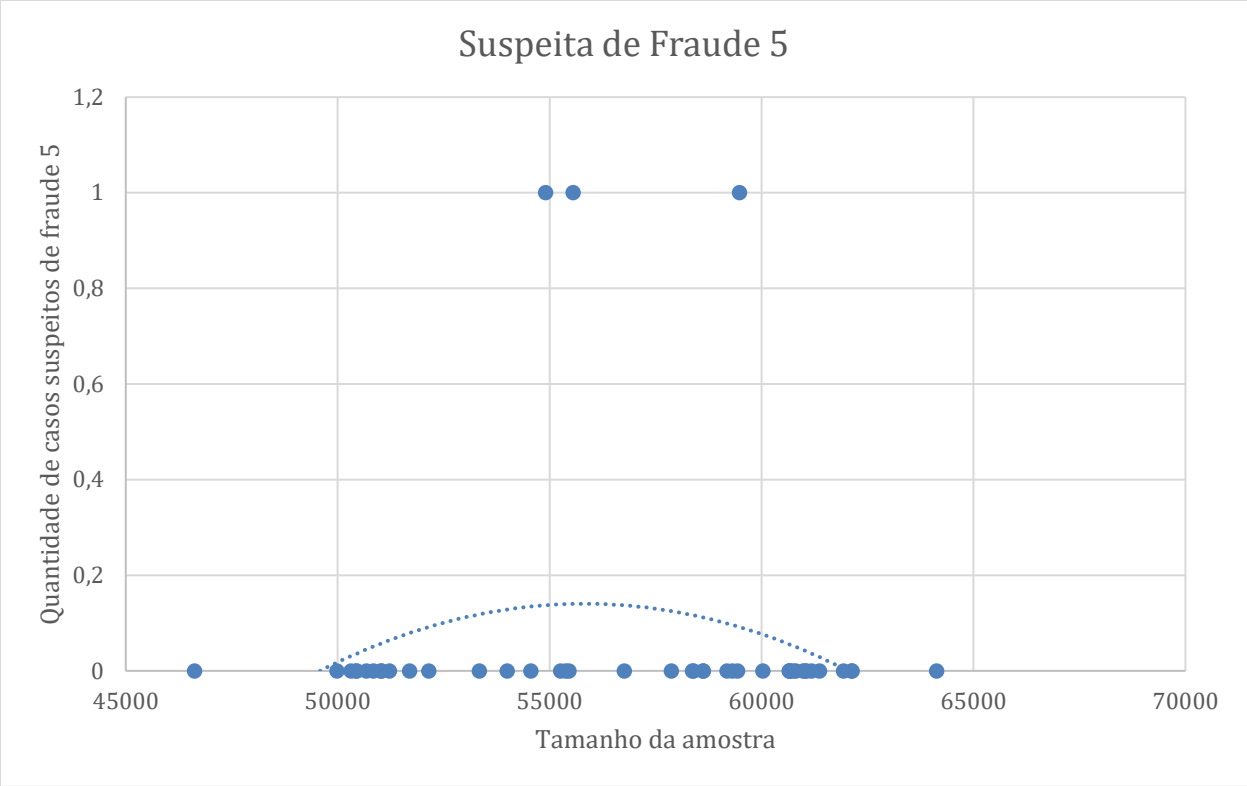


Gráfico 21 - Suspeita de Fraude 5

O gráfico 22 “Fraude 6” ilustra a relação entre o tamanho da amostra (eixo x) e o número de casos de fraude 6 (eixo y). A maioria das amostras apresenta 0 casos de fraude, mas há uma com 1 caso. O p-valor calculado é 0,752949546, indicando que não existe uma correlação estatisticamente significativa entre as duas variáveis. Ou seja, o tamanho da amostra não afeta significativamente a quantidade de casos de fraude 6 identificados. Isso ocorre porque suspeitas deste tipo são raras e quase todas as amostras não apresentam casos de fraude 6.

No decorrer do tempo, à medida que mais resultados forem obtidos, a distribuição dos casos suspeitos pode varar, o que impactará a tendência e o p-valor. Além disso, é crucial verificar se a tendência inicial se mantém, caso contrário, será necessário corrigir os casos de suspeita de fraude para manter a tendência relativamente estável num valor reduzido ou nulo, conforme ilustrado no gráfico.

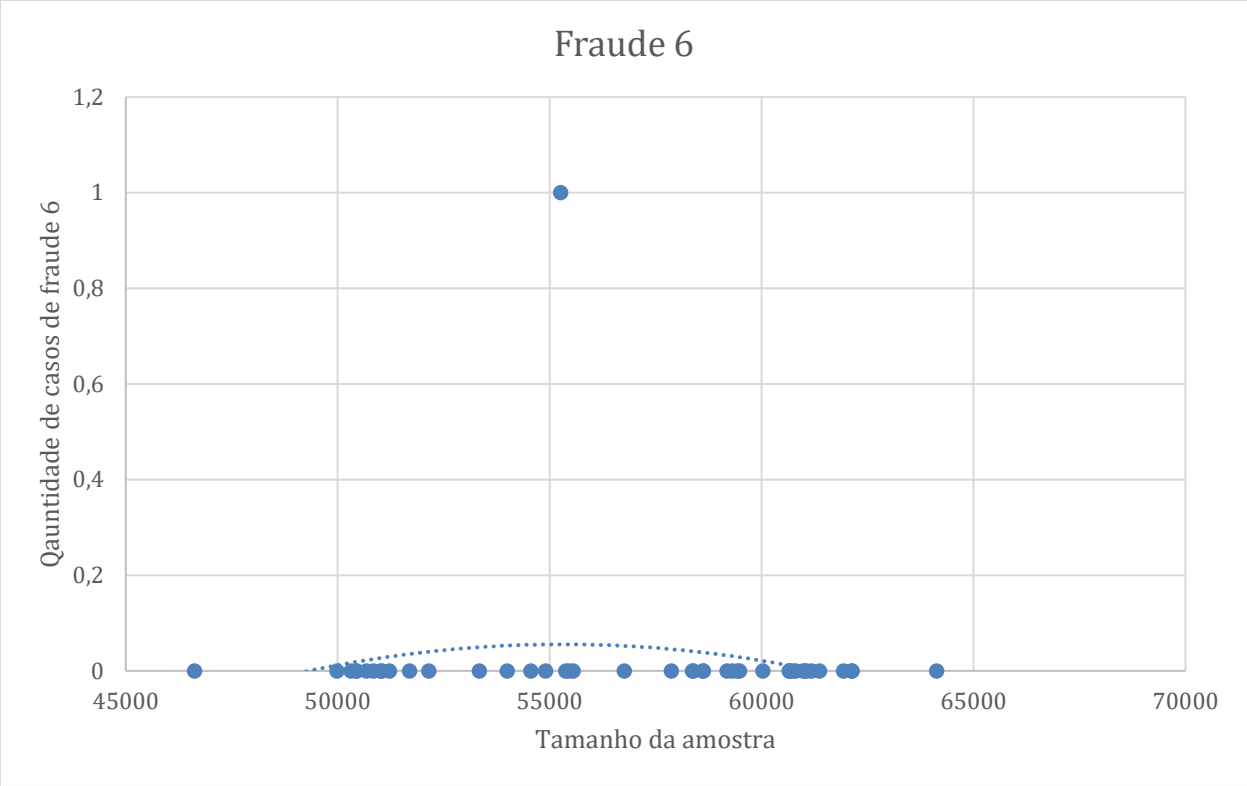


Gráfico 22 - Fraude 6

O gráfico 23 “Fraude 1”, representamos a relação entre o tamanho da amostra (eixo x) e o número de casos de fraude 1 (eixo y). Novamente, embora a maioria das amostras apresente 0 casos de fraude, existem duas amostras com 2 casos cada. O p-valor calculado é 0,231271972, indicando que o tamanho da amostra não tem um impacto substancial na quantidade de casos de fraude 1 identificados, isso ocorre porque, fraudes deste tipo são raras e quase todas as amostras não apresentam casos de fraude 1.

Contudo, no futuro à medida que mais amostras forem testadas, a distribuição dos casos de fraude pode comportar de maneira diferente, impactando a tendência e o p-valor. Além disso, é crucial verificar se esta tendência se mantém, caso contrário, será necessário corrigir os casos de fraude para manter a tendência relativamente estável e reduzida, conforme ilustrado no gráfico.

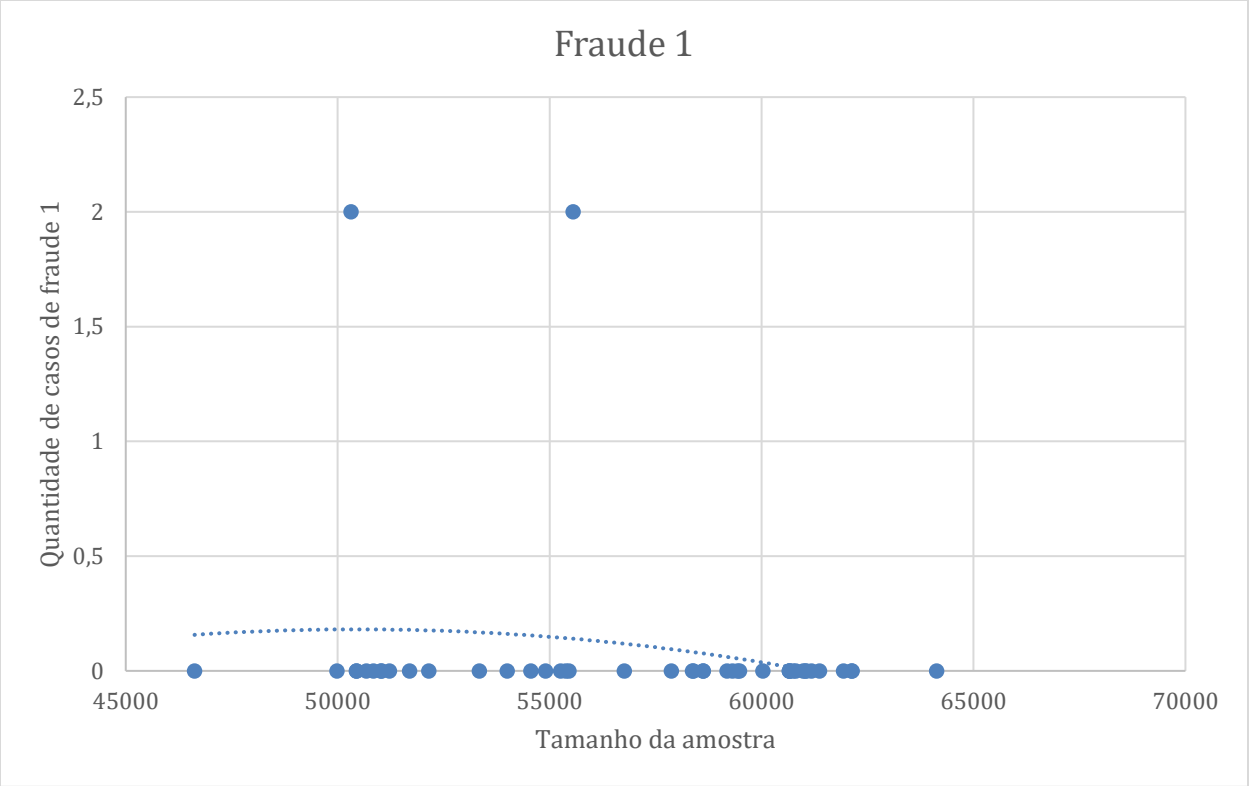


Gráfico 23 - Fraude 1