



Deteção de acessos suspeitos de utilizadores nominais em BDs

Francisco Romba António

Mestrado em Segurança Informática

Trabalho de Projeto orientado por:
Prof^a. Doutora Ana Luisa do Carmo Correia Respício
Engenheiro Ricardo Gonçalves Ramalho

Agradecimentos

A realização desta tese só foi possível graças ao apoio, orientação e incentivo de várias pessoas que, de diferentes formas, contribuíram para o sucesso deste projeto. Gostaria de deixar aqui o meu sincero agradecimento a todos eles.

Em primeiro lugar, agradeço à minha orientadora, Professora Doutora Ana Luisa do Carmo Correia Respício, pela orientação e conselhos valiosos ao longo de todo o projeto. A sua experiência, disponibilidade e rigor científico foram fundamentais não só para o desenvolvimento desta tese, como também para o meu crescimento académico e pessoal.

Agradeço também ao meu coorientador na Altice Portugal, Engenheiro Ricardo Gonçalves Ramalho, pela confiança e conhecimento partilhado, que me permitiram enfrentar os desafios do projeto com segurança.

A toda a equipa de Data Activity Monitoring and Security (DAMS), Adriana Luz, João Libório, Raúl Koch e Bruno Cravo, o meu muito obrigado pelo companheirismo, colaboração e partilha de conhecimento. Em especial, quero agradecer ao José Águas, pelo suporte técnico, pelas discussões construtivas, pela sua disponibilidade e principalmente pela amizade criada que tornou o desenvolvimento do projeto muito mais gratificante.

Aos meus colegas da equipa de User & Entity Behavior Analytics (UBA), João Neves e Madalena Guerra, agradeço a companhia, a entreaajuda e os momentos de partilha convosco, que tornaram esta experiência mais divertida e agradável.

À minha família, o meu agradecimento mais profundo: aos meus pais, irmãos, avós e restantes familiares, pelo apoio incondicional, pela confiança nas minhas capacidades e pelo incentivo constante. Sem vocês, este percurso teria sido muito mais difícil.

Por último, agradeço à minha namorada, Ilda, pelo apoio emocional, paciência e motivação, especialmente nos momentos mais exigentes do projeto. Ter-te ao meu lado tem sido uma das maiores alegrias da minha vida e um apoio essencial em todos os desafios.

A todos os mencionados e a todos aqueles que de alguma forma contribuíram para este trabalho, o meu sincero e eterno obrigado.

À amizade, à tranquilidade e aos pequenos momentos que fazem a diferença.

Resumo

A detecção de acessos suspeitos em bases de dados é fundamental para proteger informação crítica em organizações do setor das Tecnologias de Informação e Comunicação (TIC), como a MEO. Com a crescente digitalização, as bases de dados tornaram-se ativos estratégicos e, simultaneamente, alvos de ameaças. Nesse contexto, a Direção de Cibersegurança (DCY) da MEO definiu como prioridade a implementação de mecanismos de monitorização que assegurem a confidencialidade, integridade e disponibilidade da informação.

Este projeto teve como objetivo desenvolver uma solução automatizada para detetar acessos suspeitos de utilizadores nominais, combinando monitorização contínua, análise comportamental, geração de relatórios de incidentes e integração com capacidades de resposta rápida.

A solução foi estruturada em três casos de uso: (1) *HouseKeeping BD*, para identificar contas associadas a colaboradores que já saíram da empresa; (2) Partilha de Contas, destinada a detetar utilizações indevidas ou partilhas de credenciais; e (3) Monitorização de DBAs, centrada na detecção de acessos a dados críticos por administradores de bases de dados. Cada caso inclui recolha e normalização de dados, detecção de comportamentos suspeitos e geração automática de evidências e relatórios, formando um sistema completo de monitorização e alerta.

A validação de uma amostra em ambiente real permitiu obter resultados relevantes: no *HouseKeeping BD*, 13.51% das contas analisadas pertenciam a ex-colaboradores, das quais 3.11% permaneciam ativas; na partilha de contas, 99% dos casos envolveram contas aplicacionais com utilizadores nominais, com severidade entre 2.5 e 6.25; e na monitorização de DBAs, foi identificado um acesso legítimo a dados críticos corretamente sinalizado.

Os resultados confirmam a robustez e fiabilidade da abordagem, reforçando a segurança da informação, a resiliência operacional e a conformidade com o RGPD.

Palavras-chave: Bases de dados, RGPD, Acessos suspeitos, Automação

Abstract

The detection of suspicious access to databases is essential to protect critical information in organizations within the Information and Communication Technologies (ICT) sector, such as MEO. With increasing digitalization, databases have become both strategic assets and potential targets for threats. In this context, the Cybersecurity Department (DCY) at MEO has prioritized the implementation of monitoring mechanisms that ensure the confidentiality, integrity, and availability of information.

This project aimed to develop an automated solution to detect suspicious access by **nominal users**, combining continuous monitoring, behavioral analysis, incident reporting, and integration with rapid response capabilities.

The solution was structured around three use cases: (1) *HouseKeeping DB*, designed to identify database accounts associated with employees who have left the company; (2) Account Sharing, focused on detecting improper use or credential sharing; and (3) DBA Monitoring, aimed at detecting access to critical data by database administrators. Each use case includes data collection and normalization, detection of suspicious behavior, and automatic generation of evidence and reports, forming a complete monitoring and alerting system.

Validation of a real-world sample produced relevant results: in *HouseKeeping DB*, 13.51% of analyzed accounts belonged to former employees, with 3.11% still active; in Account Sharing, 99% of cases involved application accounts accessed by nominal users, with severity scores ranging from 2.5 to 6.25; and in DBA monitoring, one legitimate access to critical data was correctly flagged by the system.

The results confirm the robustness and reliability of the proposed approach, strengthening information security, operational resilience, and compliance with the RGPD.

Keywords: Databases, GDPR, Suspicious accesses, Automation

Conteúdo

Lista de Figuras	xiii
Lista de Tabelas	xv
Abreviaturas	xvii
1 Introdução	1
1.1 Enquadramento	2
1.2 Objetivo	3
1.3 Estrutura do documento	3
2 Contexto	5
2.1 Conceitos	5
2.1.1 Proteção de Dados	5
2.1.2 Tipos de Acessos Suspeitos e Ameaças Internas	7
2.1.3 Bases de Dados Relacionais	8
2.1.4 Utilizadores nominais em Bases de Dados	8
2.1.5 Conceitos de deteção de acessos suspeitos	9
2.2 Tecnologias	9
2.2.1 <i>IBM Guardium</i>	9
2.2.2 Criação e implementação de políticas	16
2.2.3 <i>ELK Stack</i>	17
2.2.4 <i>HIDRA</i>	19
2.2.5 <i>Docker</i>	19
2.2.6 <i>Tableau</i>	20
3 Trabalho Relacionado	23
3.1 PoirotDB	23
3.2 PoirotDBv2	23
3.3 TabSec	25
3.4 PID(iscover)	26

4 Fontes de Dados e Implementação nas Ferramentas	27
4.1 Fontes de Dados	27
4.2 Processamento dos dados	28
4.2.1 Normalização dos dados	29
4.2.2 Filtragem dos dados	30
4.3 IBM Guardium	31
4.3.1 Criação do grupo	31
4.3.2 Criação da política	31
4.4 Elasticsearch	32
4.4.1 Abordagens Disponíveis	32
4.4.2 Discussão das abordagens	33
4.4.3 Comparação de Desempenho: API vs SQL	33
4.4.4 Considerações Finais	34
5 Casos de Uso	35
5.1 Definição	35
5.1.1 HouseKeeping BD	35
5.1.2 Partilha de Contas	37
5.1.3 Monitorização DBAs	38
5.2 Implementação	39
5.2.1 HouseKeeping BD	39
5.2.2 Partilha de Contas	47
5.2.3 Monitorização DBAs	60
6 Conclusão e Trabalho Futuro	65
6.1 Conclusão	65
6.2 Trabalho Futuro	66
Bibliografia	73
Índice	74
A Diagrama de Gantt	75

Lista de Figuras

2.1	Arquitetura geral simplificada do <i>IBM Guardium</i> .	11
2.2	Criação de um grupo. Na primeira linha mostra-se o processo de criação do grupo dos DBAs, e nas outras linhas é apresentado outro exemplo fictício.	13
2.3	Criação de política no <i>IBM Guardium</i> .	14
2.4	Criação de política de <i>Data Security</i> no <i>IBM Guardium</i> .	14
2.5	Criação de política de <i>Session Level</i> no <i>IBM Guardium</i> .	15
2.6	Criação de política de <i>Advanced Session Level</i> no <i>IBM Guardium</i> .	15
2.7	Exemplo da criação de uma política no <i>IBM Guardium</i> .	16
2.8	Pipeline do <i>Elastic Stack</i> .	17
4.1	Processo de ETL do <i>IBM Guardium</i> para o <i>Elasticsearch</i> .	30
4.2	Criação da fonte de dados integrada no processo de criação do grupo de DBAs.	31
4.3	Criação da política de monitorização de acessos a dados críticos por DBAs (adaptado de [17]).	32
5.1	Arquitetura macro geral.	35
5.2	Arquitetura macro geral de <i>HouseKeeping BD</i> .	37
5.3	Arquitetura macro geral de Partilha de Contas.	38
5.4	Arquitetura Macro DBAs.	39
5.5	Template de reporting via <i>e-mail</i> do <i>HouseKeeping BD</i> .	42
5.6	Visão detalhada do processo de <i>HouseKeeping BD</i> .	43
5.7	<i>Dashboards</i> do processo de <i>HouseKeeping BD</i> que contém métricas como a taxa de deteção, taxa de deteção por tecnologia de BD e uma <i>timeline</i> que mostra o número de casos detetados ao longo do tempo.	46
5.8	Casos detetados no processo de <i>HouseKeeping BD</i> que continuam a realizar ações após a data de saída do colaborador da organização.	46
5.9	Cenário correto de um utilizador aplicacional aceder a uma conta de base de dados aplicacional.	49
5.10	Cenário correto de um utilizador nominal aceder a uma conta de base de dados nominal.	49
5.11	Arquitetura Micro da Partilha de Contas.	55
5.12	Acesso correspondente a partilha de conta.	57

5.13 <i>Dashboard</i> de partilha de contas.	59
5.14 Arquitetura micro da Monitorização aos DBAs.	61
5.15 <i>Dashboard</i> da monitorização de acessos dos DBAs a dados críticos. O mesmo contém métricas como o número de casos detetados, evidências dos mesmos e um gráfico temporal que mostra a evolução temporal dos casos detetados.	62
A.1 Apresentação do diagrama de Gantt cumprido ao longo da tese.	75

Lista de Tabelas

2.1 Exemplo de base de dados relacional.	8
3.1 Comparação entre os projetos PoirotDB, PoirotDB v2 e o presente trabalho.	25
3.2 Comparação entre os projetos PoirotDB, PoirotDBv2 e TabSec.	26
4.1 Exemplo de correlação de dados.	28
4.2 Comparação dos testes de desempenho entre consulta via API e consulta via SQL.	33
5.1 Fatores considerados no cálculo da severidade.	41
5.2 Combinações a reportar.	44
5.3 Combinações a não reportar.	44
5.4 Origem dos casos detetados.	45
5.5 Campos dos logs de base de dados.	48
5.6 Cenários implementados para deteção de partilha de contas.	50
5.7 Atribuição do <i>numeroUtilizadoresScore</i> .	52
5.8 Atribuição do <i>numeroLoginsScore</i> .	52
5.9 Atribuição do <i>numeroAcoesCriticasScore</i> .	53
5.10 Informação da conta de base de dados detetada em partilha.	54
5.11 Contas de rede que acederam à conta de base de dados partilhada.	54
5.12 Computadores que acederam à conta de base de dados partilhada (quando não existe informação de <i>osuser</i>).	54

Abreviaturas

ASLP Advanced Session Level Policy.

CNCS Comissão Nacional de Cibersegurança.

CNPD Comissão Nacional de Proteção de Dados.

DBA Data Base Administrator.

DCY Direção de CyberSecurity.

DSL Domain Specific Language.

DSP Data Security Policy.

ETL Extract Transform Load.

HIDRA High Performance Infrastructure for Data Research and Analysis.

HK House Keeping.

ML Machine Learning.

R2L Remote to Local.

RGPD Regulamento Geral sobre a Proteção de Dados.

SLP Session Level Policy.

SQL Structured Query Language.

U2R User to Root.

Capítulo 1

Introdução

As bases de dados são ativos críticos e fundamentais para o negócio das organizações, pois contêm informação sensível e estratégica. O termo informação sensível/crítica refere-se a dados que, se acedidos ou divulgados indevidamente, podem comprometer a privacidade, segurança ou operações de uma organização ou de um indivíduo. Refira-se alguns exemplos, os dados pessoais identificáveis (nomes, números de identificação, contactos), os detalhes financeiros (contas bancárias, cartões de crédito), os sigilos comerciais e a propriedade intelectual.

A proteção eficaz das bases de dados é essencial para garantir a confidencialidade, integridade, autenticidade e disponibilidade da informação, aspetos que constituem a base da segurança da informação. As ameaças às bases de dados são variadas e incluem acessos não autorizados, manipulação de dados, roubo de informação e até destruição intencional de dados. Estas ameaças podem resultar em danos financeiros, perda de reputação e, em casos graves, impactos legais. No contexto da segurança, uma base de dados vulnerável representa uma porta de entrada para ataques internos e externos, tornando a proteção proativa uma prioridade para qualquer organização.

É neste cenário que surge a necessidade de aplicar técnicas avançadas de monitorização e deteção de ameaças, que permitam identificar e responder rapidamente a acessos suspeitos. Estas técnicas têm o objetivo de prevenir e mitigar as consequências de ataques, protegendo assim o ativo valioso que é a informação corporativa.

O caso de violação de privacidade do Hospital do Barreiro-Montijo [14] é um exemplo marcante das consequências de falhas na proteção de bases de dados sensíveis, particularmente no setor da saúde, onde os dados tratados são considerados de alta sensibilidade.

Em 2018, uma auditoria da Comissão Nacional de Proteção de Dados (CNPd) revelou infrações graves ao RGPD, incluindo acessos indevidos a dados clínicos, perfis de utilizadores mal configurados e ausência de monitorização adequada. Foi descoberto que funcionários sem motivos profissionais tinham acesso irrestrito a informações médicas (apenas existiam 296 médicos, e haviam 985 utilizadores denominados como tal), violando o princípio da minimização de dados e expondo os pacientes e a instituição a sérios riscos de privacidade e segurança. Como consequência, foi aplicada uma multa de 400.000 euros [14], sendo a coima mais elevada imposta pela CNPD, no âmbito do RGPD, evidenciando a gravidade das falhas e a necessidade de tomar medidas robustas de segurança para garantir a confidencialidade e integridade dos dados. Este

caso demonstra a importância de soluções avançadas, como monitorização contínua e controlo rigoroso de acessos, para prevenir violações e mitigar impactos legais, financeiros e reputacionais.

1.1 Enquadramento

Os acessos ilícitos ou indevidos realizados por **utilizadores nominais** têm de ser detetados e corrigidos na MEO, pois violam o **RGPD** e as políticas de segurança da organização.

Considera-se acesso ilícito todo acesso que ocorra sem autorização formal ou em violação de legislação ou das políticas internas. Exemplos incluem a utilização de contas de colaboradores que já não pertencem à organização, o uso de contas ativas sem a devida autorização, ou acessos a dados críticos por utilizadores com permissões elevadas sem justificação. A complexidade do ambiente de bases de dados e o elevado volume de operações tornam inviável a deteção manual destes comportamentos, aumentando o risco de que acessos não autorizados passem despercebidos.

O presente trabalho foca-se na deteção de acessos suspeitos, ou seja, acessos identificados automaticamente através de regras e padrões predefinidos, que podem indicar uma violação de políticas ou uma utilização indevida. Todo acesso ilícito detetado pelo sistema é, por definição, suspeito, uma vez que viola regras ou políticas; no entanto, nem todo acesso suspeito se confirma como ilícito, pois algumas ocorrências suspeitas podem ser legítimas e devidamente autorizadas.

Os acessos inicialmente sinalizados como suspeitos passam por um processo de validação adicional, que inclui recolha de evidências detalhadas e análise contextual, para confirmar se constituem efetivamente acessos ilícitos. A deteção automática funciona assim como um primeiro filtro, permitindo priorizar os casos que requerem intervenção humana e reduzir a carga de monitorização manual.

O processo proposto realiza: (1) identificação automática de acessos suspeitos, (2) recolha de evidências detalhadas, (3) notificação automática do responsável hierárquico do utilizador envolvido, e (4) análise e correção, que constitui o momento em que se verifica quais dos acessos suspeitos são efetivamente indevidos. Esta análise pode incluir a eliminação de privilégios indevidos ou a obtenção de justificação formal quando o acesso se revelar legítimo. Os dados para esta deteção provêm de fontes internas, como bases de dados relacionais corporativas e sistemas como o **Elasticsearch**, que fornecem informações sobre contas, contexto do acesso e outras evidências necessárias para o reporte e auditoria.

Internamente, na MEO, já foram realizados alguns **trabalhos de deteção de anomalias**¹, que se focam na identificação de desvios gerais do padrão de utilização. No entanto, nenhum destes trabalhos abrange de forma completa os requisitos e objetivos específicos deste projeto. Este trabalho visa automatizar e tornar mais precisa a deteção destes acessos suspeitos, aumentando o leque de casos cobertos. A solução proposta combina tecnologias como *IBM Guardium*, *Elastic Stack* e *Python* para permitir monitorização quase em tempo real, reduzir intervenção manual e reforçar a segurança, alinhando-se com as melhores práticas e garantindo conformidade com normas como o RGPD.

¹ desvios estatísticos ou comportamentais face ao padrão normal de utilização

1.2 Objetivo

Este projeto teve como principal objetivo criar uma solução que permitisse identificar de forma automatizada acessos suspeitos de utilizadores nominais em bases de dados da MEO.

Para tal, foram realizadas as seguintes ações:

- Desenvolveram-se mecanismos de deteção de acessos suspeitos, utilizando o *IBM Guardium* e o *Elasticsearch*, com análise em tempo útil;
- Definiram-se casos de uso para identificar acessos suspeitos.

Como objetivos secundários, necessários para a concretização do objetivo principal, realizaram-se:

- Criação de *scripts*, na linguagem *Python*, capazes de detetar comportamentos suspeitos com base em padrões de acesso e regras predefinidas;
- Automatização do reporte de situações anómalas;
- Garantia de uma integração eficaz entre a monitorização contínua e a capacidade de resposta a incidentes, permitindo ações rápidas para mitigar potenciais riscos.

A concretização destes objetivos garantiu uma integração eficaz entre a monitorização contínua e a capacidade de resposta a incidentes.

1.3 Estrutura do documento

Este documento está organizado em seis capítulos principais.

- No capítulo 1: Introdução, são apresentados o enquadramento, os objetivos do projeto e a estrutura do documento;
- No capítulo 2: Contexto, são explorados os conceitos fundamentais, as tecnologias envolvidas e a integração de ferramentas como o *IBM Guardium* e o *Elasticsearch* no âmbito do projeto;
- No capítulo 3: Trabalho Relacionado, são discutidos trabalhos relacionados, incluindo projetos anteriores relevantes e abordagens existentes para a deteção de acessos suspeitos em bases de dados;
- No capítulo 4: Fontes de Dados e Implementação nas Ferramentas, apresenta e descreve as fontes de dados utilizadas para a realização dos casos de uso, bem como o trabalho desenvolvido nas ferramentas *Elasticsearch* e *IBM Guardium*, detalhando os passos de implementação.

- No capítulo 5: Casos de Uso, explica o processo de definição dos casos de uso, incluindo os respectivos diagramas macro que ilustram a sua conceção, e descreve em detalhe a implementação prática de cada caso de uso definido.
- Por fim, o capítulo 6: Conclusão e Trabalho Futuro, apresenta as principais conclusões do projeto e propõe possíveis direções para trabalhos futuros.

Capítulo 2

Contexto

Neste capítulo, pretende-se clarificar alguns termos e conceitos essenciais para a compreensão do trabalho. Na secção 2.1, serão abordados conceitos gerais relevantes para o tema, enquanto na secção 2.2 serão explicadas e correlacionadas algumas das tecnologias que serão utilizadas ao longo do projeto.

2.1 Conceitos

Neste capítulo serão apresentados conceitos fundamentais para o desenvolvimento do trabalho. São abordados tópicos que fornecem o enquadramento necessário para a compreensão das questões exploradas ao longo do trabalho, destacando aspetos cruciais como proteção de dados e regulamentações associadas.

2.1.1 Proteção de Dados

A proteção de dados refere-se ao conjunto de práticas, políticas e tecnologias que garantem a confidencialidade, integridade e disponibilidade da informação. Num contexto organizacional, este conceito assume especial relevância, pois está diretamente ligado à salvaguarda de dados sensíveis, como informações pessoais, financeiras e empresariais. A proteção de dados não é apenas uma boa prática, mas também uma exigência legal e ética, sendo fundamental para mitigar riscos de acessos indevidos, roubo de informação e outros incidentes de segurança.

Regulamento Geral sobre a Proteção de Dados

O RGPD é uma legislação europeia em vigor desde 25 de maio de 2018 [8], que tem como objetivo reforçar a proteção de dados pessoais dos cidadãos da União Europeia.

Em Portugal, a entidade responsável por supervisionar a aplicação do RGPD é a CNPD.

Comissão Nacional de Proteção de Dados

A CNPD é uma entidade administrativa independente portuguesa, com personalidade jurídica de direito público e autonomia administrativa e financeira, que atua junto da Assembleia da República. É responsável por controlar e fiscalizar o cumprimento do RGPD e das leis nacionais relacionadas com a proteção de dados pessoais, visando garantir os direitos, liberdades e

garantias das pessoas singulares no tratamento de seus dados[6].

Política de Segurança da Informação do Grupo MEO

No caso específico do Grupo MEO, a Política de Segurança da Informação [18] estabelece diretrizes e práticas para proteger os dados sensíveis e assegurar a conformidade com os requisitos legais e regulatórios aplicáveis. Esta política visa garantir que todos os dados e sistemas da organização são adequadamente protegidos contra acessos não autorizados, perdas ou compromissos, promovendo um ambiente de segurança robusto e confiável.

A Política de Segurança da Informação da MEO define, entre outros aspetos, as responsabilidades de cada utilizador no manuseamento seguro de dados e descreve as medidas de segurança a serem seguidas, como o controlo de acessos, a gestão de permissões e a monitorização de atividades suspeitas; Também especifica os processos de resposta a incidentes, essenciais para a rápida identificação e mitigação de ameaças.

No contexto deste trabalho, a Política de Segurança da Informação da MEO serve como um quadro de referência para a implementação de práticas seguras de acesso e monitorização de bases de dados, assegurando que o trabalho desenvolvido está alinhado com as exigências de segurança da organização e com melhores práticas do setor.

De seguida, listam-se algumas regras referidas no documento [18] relacionados com o tema:

- “Pedidos de novos Utilizadores ou de alteração de privilégios devem passar por um processo de validação que inclua a área de Recursos Humanos, no caso dos internos e das áreas responsáveis pela contratação, no caso dos externos, devendo ser efetuados por escrito, utilizando templates pré-definidos ou aplicações informáticas específicas para o efeito e aprovados pela respetiva chefia antes de implementados”;
- “Todos os Utilizadores inativos durante 60 dias (máximo) deverão ser automaticamente bloqueados; Utilizadores inativos durante 1 ano (máximo) deverão, se possível tecnicamente e sem impacto na rastreabilidade dos Utilizadores, ser automaticamente removidos, caso contrário deverão ser mantidos bloqueados”;
- “Privilégios atribuídos a Utilizadores externos à organização deverão expirar automaticamente, no máximo, ao fim de 90 dias. Exceções a esta regra poderão ser implementadas, desde que sejam identificadas pelo respetivo interlocutor autorizado da MEO Portugal, que definirá o prazo adequado para o respetivo acesso. Sempre que seja necessário prolongar este período deverá ser autorizado novo pedido, pela mesma entidade”;
- “Sempre que exista uma alteração de responsabilidades ou funções de um Utilizador a respetiva chefia deve solicitar a correspondente adequação das contas de acesso do Utilizador”;
- “Para sistemas e aplicações críticas, é imperativa a execução de um procedimento que obrigue periodicamente à revalidação, por parte dos responsáveis hierárquicos, a necessidade da manutenção do acesso aos Utilizadores, essa revalidação deverá acontecer no mínimo com uma periodicidade anual para cada utilizador”.

2.1.2 Tipos de Acessos Suspeitos e Ameaças Internas

A identificação de acessos suspeitos consiste em reconhecer padrões de acesso que fogem ao comportamento esperado de um utilizador legítimo, ou seja, comportamentos anómalos. Estes acessos podem estar relacionados a ações de abuso de privilégios, em que um utilizador comete ações além daquilo que lhe é permitido, ou ao uso de credenciais comprometidas, como quando um atacante obtém acesso a uma conta de um utilizador interno.

Um dos maiores desafios de segurança são as ameaças internas, que, de acordo com o Centro Nacional de Cibersegurança (CNCS) “traduz-se no abuso de forma involuntária ou intencional, de qualquer atual colaborador ou ex-colaborador, sócio ou fornecedor, que tenha, ou tenha tido, acesso aos ativos digitais da organização” [7].

Dentro das ameaças internas, existem três tipos de agentes internos [10]:

- **Maliciosos:** geralmente são colaboradores atuais ou ex-colaboradores que estão insatisfeitos cujas credenciais de acesso não foram retiradas/eliminadas, e assim usam o seu acesso de maneira intencional por vingança, ganho financeiro ou outros motivos mal intencionados. Ataques relacionados com este tipo de agente aconteceram recentemente [10]:
 - No início da pandemia de COVID-19, um ex-funcionário de uma empresa de embalagens médicas usou uma conta de administrador criada anteriormente para configurar uma nova conta de utilizador falsa. Depois alterou milhares de arquivos de forma a atrasar ou interromper o envio de equipamentos de proteção individual para hospitais e prestadores de serviços de saúde [11];
 - Em 2022, um funcionário do X foi preso por enviar informações privadas de utilizadores da plataforma para fora [12].
- **Negligente:** colaboradores que não têm intenção maliciosa, mas são uma vulnerabilidade/ameaça devido à sua ignorância ou descuido, por exemplo, ser vítima de um ataque de *phishing*¹, ignorar controlos de segurança estabelecidos pela empresa, perder o computador dado pela empresa, deixar credenciais de acesso expostas a pessoas de fora, entre outros descuidos;
- **Comprometido:** utilizadores legítimos cujas credenciais foram comprometidas. As ameaças internas via utilizadores comprometidos são as ameaças internas mais caras. Por exemplo, em 2021, um golpista usou uma tática de engenharia social² (especificamente uma chamada telefónica de *phishing* de voz, ou *vishing*³) para conseguir as credenciais de acesso aos sistemas de atendimento ao cliente na plataforma de comércio *Robinhood*. Mais de 5 milhões de endereços de e-mail de clientes e 2 milhões de nomes de clientes foram roubados no ataque [19].

¹técnica de fraude que usa mensagens enganosas para obter dados sensíveis.

²<https://www.ibm.com/br-pt/topics/social-engineering>

³técnica de fraude que usa chamadas telefónicas para obter dados sensíveis.

Estas ameaças representam uma proporção significativa dos incidentes de segurança em empresas. Os colaboradores são um dos principais pilares dentro de uma empresa contudo são aqueles que apresentam um maior risco/vulnerabilidade à mesma.

Esses acessos suspeitos podem envolver ações como consultas massivas a dados (indicando uma possível exfiltração), tentativas de acesso a dados fora do horário normal ou aceder a tabelas de dados confidenciais que o utilizador não costuma aceder/não costuma ter privilégios para o fazer. O problema é agravado pela dificuldade em detetar anomalias numa base de dados, já que os utilizadores internos têm permissões e acessos rotineiros e as suas ações podem ser difíceis de diferenciar face a comportamentos maliciosos.

2.1.3 Bases de Dados Relacionais

Uma base de dados relacional, é um sistema para armazenar e organizar dados usando tabelas que se relacionam entre si. Cada tabela é como uma folha de cálculo, com linhas e colunas que definem as informações. Por exemplo, uma tabela pode guardar informações de clientes, e outra, os pedidos deles.

Essas tabelas podem-se conectar por meio de chaves (IDs), criando relações entre os dados, como “este cliente fez estes pedidos”. Essa estrutura é baseada na lógica relacional, o que facilita a consulta, manipulação e garantia de consistência dos dados.

Bases de dados relacionais usam linguagens como a *Structured Query Language* (SQL) para interagir com os dados e são ideais para aplicações que precisam de estrutura organizada, integridade e transações confiáveis.

Na Tabela 2.1, mostra-se um exemplo de uma base de dados constituída apenas por duas tabelas: clientes e pedidos.

Tabela 2.1: Exemplo de base de dados relacional.

Tabela dos clientes		Tabela dos pedidos		
ID	Nome	ID	Cliente	Produto
1	Maria	1	1	TV
2	Eduardo	2	1	Telemóvel
3	Jorge	3	2	Tablet

2.1.4 Utilizadores nominais em Bases de Dados

No contexto da segurança de bases de dados, um utilizador nominal refere-se a um utilizador padrão ou comum, que possui permissões adequadas ao seu cargo e que utiliza a base de dados de forma legítima, mas limitada ao necessário para as suas funções. Em geral, estes utilizadores possuem um conjunto de permissões que lhes permite aceder e manipular dados específicos, sem privilégios administrativos.

A monitorização de utilizadores nominais é importante porque, se as suas credenciais forem comprometidas, os atacantes podem aceder a dados sensíveis sem levantar alertas evidentes, uma vez que as suas ações podem parecer legítimas à primeira vista. Além disso, um utilizador nominal, seja por negligência ou por ação maliciosa, pode abusar das suas permissões para aceder ou manipular informação além das suas responsabilidades. No entanto, a deteção de acessos suspeitos por parte destes utilizadores é desafiadora, uma vez que a definição de um comportamento “normal” pode ser variável.

2.1.5 Conceitos de deteção de acessos suspeitos

A deteção de acessos suspeitos é o processo de identificar padrões ou acessos que violam regras ou políticas previamente definidas, podendo indicar potencial acesso ilícito. Esta abordagem é fundamental para a segurança de bases de dados, pois permite distinguir entre atividades legítimas e potenciais ilícitudes de forma automatizada.

No contexto da cibersegurança, os algoritmos baseados em regras (*rule-based algorithms*) [20] aplicam um conjunto pré-definido de condições lógicas a cada evento ou fluxo de tráfego monitorizado, sinalizando como suspeito todo o acesso que corresponda a padrões de violação dessas regras. Ao contrário das abordagens de *Machine Learning* (ML), que aprendem padrões estatísticos a partir de grandes volumes de dados, os algoritmos baseados em regras não dependem de técnicas de aprendizagem, permitindo deteção direta de acessos que correspondam a cenários de risco previamente conhecidos.

Na segurança de bases de dados, as abordagens baseadas em regras são populares pela sua simplicidade e facilidade de implementação, uma vez que não exigem grandes volumes de dados para serem eficazes. As regras podem ser ajustadas manualmente para refletir padrões específicos de segurança ou conformidade, tornando esta técnica particularmente útil em ambientes onde o comportamento esperado é bem compreendido.

Ao identificar acessos suspeitos em tempo real, estes sistemas permitem uma resposta rápida e ajudam a mitigar danos que um potencial acesso indevido possa causar. Contudo, a implementação de algoritmos baseados em regras na segurança de bases de dados apresenta desafios, como a necessidade de atualização constante das regras para acompanhar novas ameaças e a possibilidade de gerar falsos positivos quando um comportamento incomum é, na verdade, legítimo.

2.2 Tecnologias

2.2.1 IBM Guardium

O *IBM Guardium*⁴ é uma ferramenta de segurança especializada na proteção e monitorização de bases de dados. A mesma auxilia a identificar, auditar e proteger dados sensíveis, permitindo que as organizações detetem acessos não autorizados e atividades suspeitas em tempo real. O *IBM Guardium* oferece funcionalidades como a monitorização contínua de acessos, a produção de

⁴<https://www.ibm.com/guardium>

relatórios de auditoria e a aplicação de políticas de segurança para garantir a conformidade com regulamentos como o RGPD. A ferramenta também pode detetar vulnerabilidades e comportamentos anómalos, ajudando a mitigar riscos relacionados com dados sensíveis.

Componentes da Arquitetura

A arquitetura do *IBM Guardium* é hierárquica e é constituída por três tipos de appliances⁵:

Coletores: É responsável por receber os dados capturados pelos agentes S-TAP instalados nos servidores de bases de dados. Estes dados incluem atividade de utilizadores, consultas SQL, alterações e acessos a dados sensíveis. Após a receção, o coletor aplica as políticas de segurança configuradas, deteta potenciais acessos indevidos, gera alertas e armazena temporariamente os registos de auditoria. Este componente pode operar de forma autónoma ou integrado numa arquitetura maior com agregadores.

Agregadores: Tem como função principal consolidar e centralizar os dados provenientes de múltiplos coletores. Ao receber grandes volumes de registos, este componente executa processos de normalização, deduplicação e compressão da informação, otimizando o desempenho da análise e reduzindo redundâncias. O agregador é essencial em ambientes empresariais com várias fontes de dados, permitindo relatórios agregados e uma visão consolidada da infraestrutura.

Gerenciador central: Atua como o ponto de controlo central da plataforma. Através desta *appliance*, os administradores gerem políticas de segurança, configurações, permissões de utilizadores e monitorizam o estado geral dos componentes do sistema. O gerenciador central distribui automaticamente as políticas para os restantes *appliances*, coordena tarefas como *backups* e atualizações, e disponibiliza dashboards e relatórios com uma visão global da atividade monitorizada. Normalmente, esta funcionalidade está integrada numa *appliance* agregador com capacidades acrescidas de gestão.

Adicionalmente, na base da arquitetura encontram-se os *data sources*, como bases de dados relacionais, sistemas de ficheiros ou aplicações empresariais. É nestes sistemas que ocorre a atividade a ser monitorizada. Para capturar essa atividade em tempo real, o *IBM Guardium* utiliza o componente S-TAP, um agente leve instalado no servidor onde reside o *data source*. Este agente intercepta todas as comunicações entre os utilizadores e a base de dados, sem afetar o seu desempenho, e envia os dados recolhidos diretamente para o coletor.

O S-TAP é, portanto, o primeiro ponto de contacto do *IBM Guardium* com os dados, sendo responsável por fornecer visibilidade completa sobre os acessos, operações e padrões de utilização. Em ambientes *cloud* ou *containers*, versões alternativas como o S-TAP externo ou os conectores universais que são utilizados para obter os dados através de *logs*, APIs ou tráfego de rede.

Fluxo cronológico dos dados

O funcionamento do *IBM Guardium* segue um fluxo sequencial que acompanha o ciclo de vida da informação monitorizada. A atividade inicia-se nos *data sources*, como bases de dados ou

⁵Sistema pré-configurado (físico ou virtual) que executa funções no *IBM Guardium*.

aplicações, onde os acessos e operações realizadas pelos utilizadores são capturados em tempo real pelo agente S-TAP.

De seguida, essa informação é transmitida para o coletor, que regista os eventos, aplica as políticas de segurança definidas e, caso necessário, gera alertas imediatos.

Quando o ambiente possui múltiplos coletores, os registos são posteriormente consolidados num agregador, que centraliza e otimiza os dados recebidos de diferentes fontes, reduzindo redundâncias e preparando-os para análises mais abrangentes.

Por fim, o gerenciador central integra toda a informação da infraestrutura, disponibilizando *dashboards*, relatórios e mecanismos de administração unificada. Esta visão centralizada permite às equipas de segurança acompanhar a atividade em tempo real, identificar anomalias e responder de forma coordenada a potenciais incidentes.

A Figura 2.1 apresenta de forma simplificada a arquitetura geral do *IBM Guardium*, ilustrando os principais componentes e o percurso cronológico dos dados descrito acima.

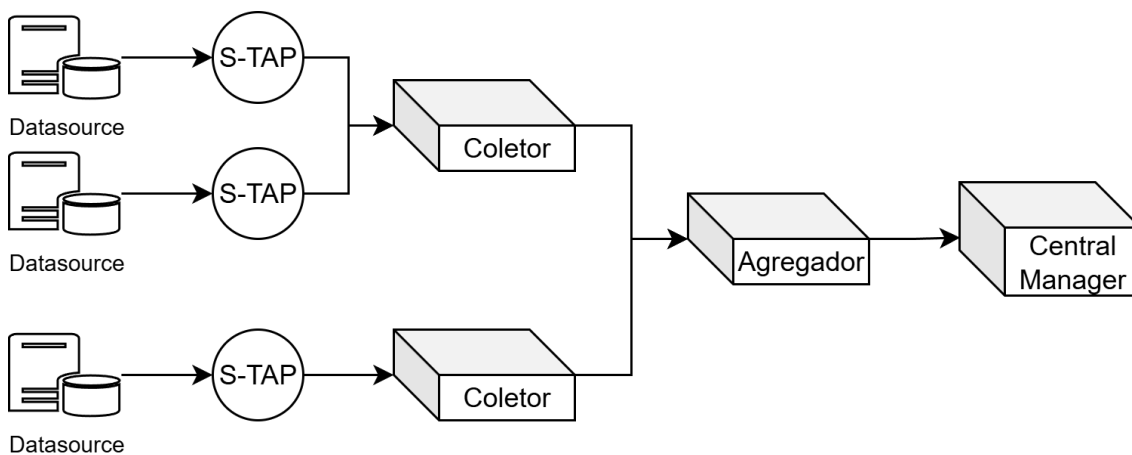


Figura 2.1: Arquitetura geral simplificada do *IBM Guardium*.

Funcionalidades

O *IBM Guardium* disponibiliza diversas funcionalidades, cada uma voltada para diferentes aspetos da segurança e gestão de dados. Neste trabalho, foram abordadas apenas três delas, por serem as que se relacionam diretamente com o âmbito do estudo: Descoberta e classificação de dados, políticas e grupos.

Descoberta e classificação de dados

Este módulo é responsável por identificar e classificar dados armazenados nas bases de dados monitorizadas pelo *IBM Guardium*. Esta funcionalidade automatiza a deteção de informações, atribuindo classificações que permitem aplicar políticas de segurança adequadas e garantir conformidade com normas e regulamentos.

Grupos

Os grupos são conjuntos de utilizadores ou entidades que compartilham características ou permissões semelhantes. Estes grupos permitem organizar as contas de utilizadores de forma mais eficiente, o que facilita a aplicação de políticas de segurança e a monitorização de acessos. Por exemplo, pode-se criar um grupo de utilizadores que tenham permissões específicas para aceder a certos tipos de dados, e as ações desses utilizadores serão monitorizadas de acordo com as regras estabelecidas.

Criação de grupos

A criação de um grupo dinâmico com base em dados extraídos por uma *query* personalizada no *IBM Guardium* requer a utilização de diversos recursos da plataforma, nomeadamente o *Custom Table Builder*, *Create Domain Builder*, *Create Query-Report Builder* e *Group Builder*. Esta estrutura modular visa garantir a reutilização, segurança e consistência dos dados em todo o sistema, sendo por isso necessário seguir uma sequência bem definida de passos.

Os passos são os seguintes:

- **Criação de uma entidade no módulo "Custom Table Builder"**: esta entidade define a estrutura de dados esperada, incluindo o nome da tabela lógica e a *query* SQL utilizada para extrair os dados relevantes. Esta camada permite validar e organizar os dados antes da sua utilização noutros componentes;
- **Criação de um domínio no módulo "Create Domain Builder"**: o domínio agrupa uma ou mais entidades, permitindo controlar o seu contexto de utilização e a sua visibilidade dentro da plataforma. É uma camada de abstração essencial para reutilização e gestão de permissões;
- **Criação de uma *query-report* no módulo "Create Query-Report Builder"**: aqui define-se a lógica de extração dos dados específicos que se pretende utilizar. A *query-report* atua como uma fonte dinâmica que irá alimentar o grupo, permitindo aplicar filtros e selecionar colunas relevantes;
- **Criação de um grupo no módulo "Group Builder"**: finalmente, é criado o grupo propriamente dito, definindo-se o tipo de entidade (utilizadores, IPs, comandos, etc.) e a forma de preenchimento, neste caso, com base nos dados extraídos pela *query-report*. É ainda possível agendar atualizações automáticas do grupo através da funcionalidade "Schedule".

Segue abaixo a Figura 2.2 que ilustra a criação de um grupo no *IBM Guardium*, onde a linha em destaque corresponde ao processo de criação do grupo de DBAs, e as restantes linhas correspondem a um exemplo que demonstra algumas peculiaridades deste processo, como a criação de um domínio poder conter uma ou mais entidades, e o mesmo domínio poder ser "dividido" em duas *query-report's* diferentes. O mesmo processo poderia ser feito de forma diferente, onde seria

apenas criada uma *query-report* que iria buscar os dados todos do domínio, e a filtração do tipo de utilizador (nominal ou aplicativo), ser feita apenas na criação do grupo.

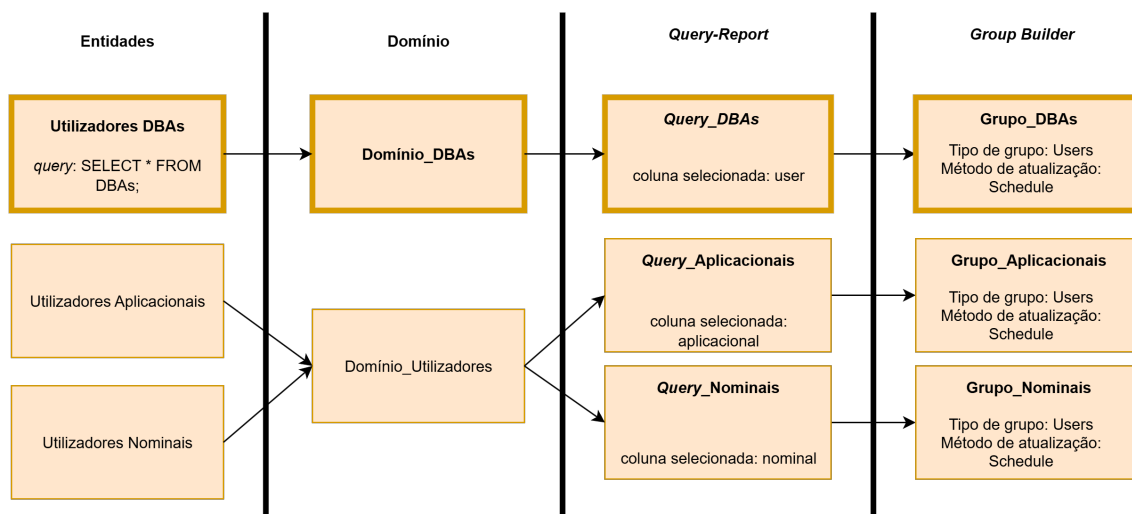


Figura 2.2: Criação de um grupo. Na primeira linha mostra-se o processo de criação do grupo dos DBAs, e nas outras linhas é apresentado outro exemplo fictício.

Este processo, embora composto por várias etapas, é fundamental para garantir que os grupos dinâmicos se mantenham atualizados e corretamente integrados com as políticas de segurança definidas no *IBM Guardium*. Para assegurar que essa atualização ocorra de forma contínua e sem intervenção manual, foi implementado um agendamento que executa automaticamente todas as etapas descritas, realizando a atualização dos grupos em intervalos definidos. Desta forma, garante-se que os grupos refletem sempre o estado mais recente dos dados, preservando a eficácia das políticas aplicadas.

Políticas

As políticas são regras configuráveis que determinam o comportamento e as permissões dos utilizadores e grupos. Existem três tipos de políticas[1]:

- **Data security policy (DSP):** Oferece o conjunto mais amplo de opções para definir regras com base em diversos critérios (como *user*, objeto, comando, horário) e ações (como alertas, registos e bloqueios). De acordo com a *IBM Guardium*, é o tipo de política mais completo, permitindo uma grande variedade de combinações entre critérios e ações;
- **Session level policy (SLP):** Projetado para validar critérios no início de uma sessão de banco de dados; Melhora o desempenho ao filtrar rapidamente o tráfego irrelevante. Permite apenas critérios e ações no nível da sessão.
- **Advanced session level policy (ASLP):** Estende as políticas de nível de sessão ao permitir regras mais complexas baseadas em sessão por meio de *scripts* usando a linguagem SR.

Segundo a *IBM Guardium*, esse tipo de política possibilita a definição de regras sofisticadas para sessões, utilizando *scripts*.

Na Figura 2.3 apresentam-se as opções de configuração necessárias para a criação de uma política.

Create New Policy

Name and properties *Provide a unique name and optional properties* [Collapse](#)

* Type ☒ Data security policy ☐ Session level policy ☐ Advanced session level policy

* Name

Category [?](#)

[Roles](#) No roles assigned.

[Show advanced options](#)

☒ Rules *Number of rules: 0* [Expand](#)

Figura 2.3: Criação de política no *IBM Guardium*.

A teoria e o processo de criação associados aos diferentes tipos de políticas variam consoante a sua natureza. No caso de uma política do tipo DSP, a secção "Nome e propriedades" permite definir um nome e uma categoria, que servirá como etiqueta para efeitos de reporte. Na definição das regras que compõem a política, é possível escolher o tipo (por exemplo, acesso, exceção, extrusão, entre outros), bem como configurar o respetivo nome, categoria, classificação e nível de severidade.

Create New Rule

Rule definition *Specify rule name and type* [Collapse](#)

Rule type

* Rule name

Category [?](#)

Classification [?](#)

Severity

Tags [+](#)

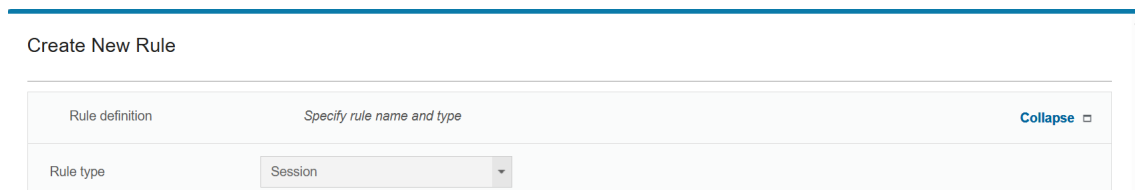
☒ Rule criteria *No parameters specified* [Expand](#)

Rule action *Number of actions: 0* [Expand](#)

Figura 2.4: Criação de política de *Data Security* no *IBM Guardium*.

Nas políticas do tipo SLP, a secção inicial limita-se à definição do nome, já que a categoria é, por definição, sempre associada a sessões. Relativamente às regras, apenas é permitido criar

regras de tipo sessão, embora as restantes propriedades (como nome, classificação e severidade) se mantenham semelhantes às da política DSP.



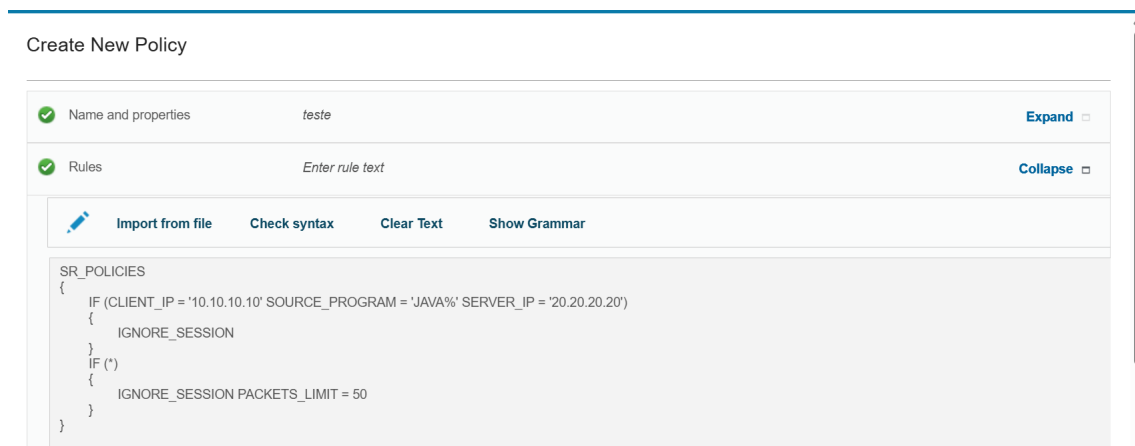
Create New Rule

Rule definition Specify rule name and type Collapse

Rule type Session

Figura 2.5: Criação de política de *Session Level* no *IBM Guardium*.

A política do tipo ASLP partilha a mesma limitação inicial quanto à categoria, também restrita ao nível de sessão. Contudo, distingue-se na forma como as regras são definidas: ao invés de seleccionar tipos predefinidos, estas políticas requerem a criação de regras através de *scripts* utilizando a linguagem SR[2], dando a possibilidade da criação de uma regra mais complexa e específica. Na Figura 2.6 é possível ver um exemplo de um *script* em linguagem SR.



Create New Policy

Name and properties teste Expand

Rules Enter rule text Collapse

Import from file Check syntax Clear Text Show Grammar

```
SR_POLICIES
{
  IF (CLIENT_IP = '10.10.10.10' SOURCE_PROGRAM = 'JAVA%' SERVER_IP = '20.20.20.20')
  {
    IGNORE_SESSION
  }
  IF (*)
  {
    IGNORE_SESSION_PACKETS_LIMIT = 50
  }
}
```

Figura 2.6: Criação de política de *Advanced Session Level* no *IBM Guardium*.

As políticas desempenham um papel essencial na segurança e conformidade do sistema, podendo ser aplicadas a diferentes níveis para controlar o acesso a dados sensíveis, monitorizar atividades e reagir a comportamentos potencialmente maliciosos. Cada tipo de política oferece diferentes graus de complexidade e flexibilidade, adaptando-se a diversos contextos e necessidades.

Ao permitir ações como bloqueios, geração de alertas ou registo de eventos, estas políticas contribuem significativamente para a proteção dos dados, garantindo o cumprimento de normas e facilitando uma resposta rápida a incidentes de segurança. Além disso, possibilitam verificar, por exemplo, se um utilizador acedeu indevidamente a informação confidencial ou se violou políticas de acesso previamente definidas. Desta forma, o uso eficaz de políticas no *IBM Guardium* constitui um componente fundamental para uma gestão de segurança robusta e proativa.

2.2.2 Criação e implementação de políticas

A criação de uma política envolve várias fases, desde a definição dos critérios de monitorização até à ativação da política no ambiente de produção. No caso concreto deste trabalho, foi criada uma política com base num grupo dinâmico de utilizadores (definido anteriormente), com o objetivo de gerar alertas sempre que esses utilizadores realizassem uma consulta a dados críticos.

Os passos realizados foram os seguintes:

- **Definição da Política (*Policy Builder*):** foi criada uma nova política no recurso *Policy Builder*, com um nome identificativo (ex: *Politica_DBAS_Acessos_Sensíveis*). A política foi configurada para monitorizar operações de consulta a dados críticas por parte de DBAs, neste caso todos os comandos que não pertençam ao grupo de comandos administrativos. Este grupo inclui, por exemplo, comandos como GRANT, ALTER e CREATE, executados por utilizadores pertencentes ao grupo *Grupo_DBAS*.
- **Criação das Regras (*Rules*):** dentro da política foram definidas regras com condições específicas. Por exemplo, uma regra com a condição: "Se o *dbuser* pertence ao grupo *Grupo_DBAS*" e a operação for SELECT, então gera um alerta;
- **Associação de Ações (*Actions*):** para cada regra foram associadas ações, como o registo de um alerta, envio de e-mail, ou integração com um SIEM externo. No contexto deste trabalho, foi configurada a geração de alertas enviados para o *ElasticSearch* e via e-mail;
- **Ativação e *Deployment* da Política:** após a configuração, a política foi ativada e enviada para os *appliances* relevantes através da opção *Install Changes*. Esta etapa é essencial para que a política passe do estado de rascunho para o estado ativo e executável.

A Figura 2.7 mostra um exemplo de como é feita uma política.

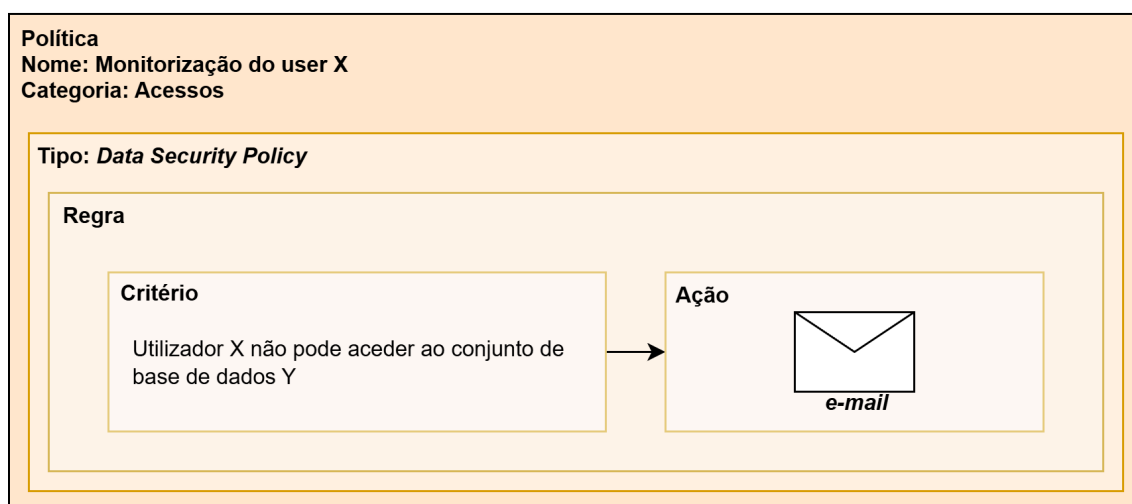


Figura 2.7: Exemplo da criação de uma política no *IBM Guardium*.

Este processo mostra a flexibilidade da ferramenta no controlo de acessos e atividades em sistemas de informação sensíveis, permitindo configurar políticas altamente específicas baseadas em dados dinâmicos.

2.2.3 ELK Stack

O *Elastic Stack* ^[6] também conhecido como *ELK Stack*, é um conjunto de ferramentas desenvolvido pela *Elastic* que permite a ingestão, armazenamento, pesquisa, análise e visualização de grandes volumes de dados em tempo quase real. O *Stack* é composto por quatro componentes principais: *Elasticsearch*, *Logstash*, *Kibana* e, mais recentemente, o *Beats*. A arquitetura modular ^[7] permite que cada componente execute uma função específica dentro de um pipeline de dados, facilitando a monitorização de sistemas, análise de *logs*, deteção de anomalias, entre outros casos de uso.

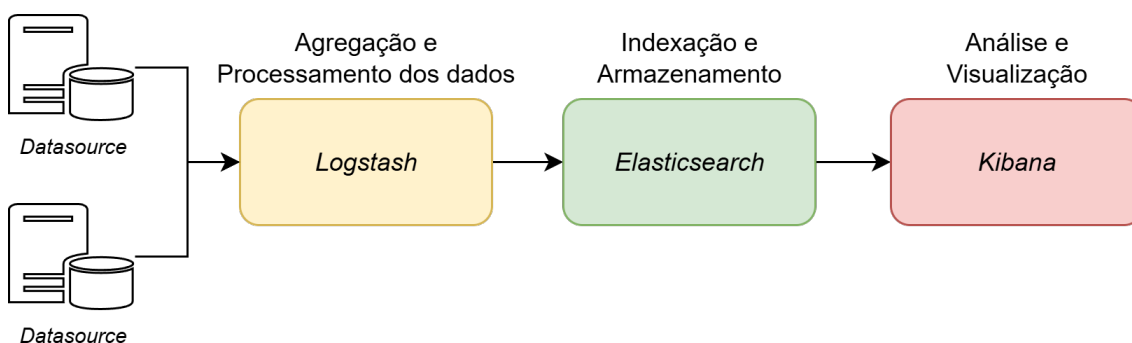


Figura 2.8: Pipeline do *Elastic Stack*.

Elasticsearch: Sistema responsável pelo armazenamento, busca e análise de dados escalável, visto que mantém a eficiência com o aumento de volume de dados. É usado para pesquisar, indexar, armazenar e analisar dados de todas as formas e tamanhos quase em tempo real. Alguns conceitos-chave do *Elasticsearch* ^[4] incluem:

- **Índice:** É um *namespace* lógico que contém uma coleção de documentos relacionados. Semelhante a uma base de dados numa arquitetura relacional, mas com a diferença de que os dados são armazenados em formato *JSON* sem esquema rígido, permitindo maior flexibilidade e eficiência na pesquisa;
- **Documento:** Unidade básica de informação armazenada num índice. Representa uma entidade do mundo real, estruturada como um objeto *JSON*;
- **Campo:** Um par chave-valor dentro de um documento, que representa uma característica específica. Por exemplo: "IP" : "10.0.0.1";

⁶<https://www.elastic.co/elastic-stack>

⁷Sistema dividido em módulos independentes que podem ser alterados ou reutilizados sem impactar o todo.

- **Namespace:** Pode ser interpretado como uma “caixa rotulada”, onde se agrupam entidades com nomes semelhantes, evitando conflitos de identificação entre índices ou tipos de dados distintos.

Elasticsearch API O acesso ao *Elasticsearch* é feito através de uma API *RESTful*⁸, que permite realizar operações como indexação de dados, pesquisa, agregações e gestão de índices. A *Search API*, em particular, permite consultar documentos com elevado nível de personalização, usando o *Query DSL (Domain Specific Language)*, permitindo filtros, pesquisas por relevância, condições booleanas, entre outros.

Kibana: Interface gráfica da *Stack*, permitindo a visualização e exploração de dados armazenados no *Elasticsearch*. Os utilizadores podem criar dashboards personalizados, gráficos, mapas, e painéis interativos com base nos dados indexados. Por exemplo, é possível gerar um gráfico com o *dbuser* mais utilizado dentro de um determinado intervalo temporal, facilitando a deteção de padrões de acesso ou comportamentos anómalos.

Logstash: Pipeline de processamento de dados que permite ingerir dados de múltiplas fontes, transformá-los conforme necessário (por exemplo, aplicar filtros, enriquecer campos, ou reestruturar o conteúdo) e encaminhá-los para destinos como o *Elasticsearch*. O processo é composto por três fases principais: *input*, *filter* e *output*, oferecendo grande flexibilidade na manipulação de dados em tempo real. Este processo é também chamado por *Export, Transform and Load (ETL)*, e este é a ponte entre o *IBM Guardium* e o *Elasticsearch*, visto que este exporta do *IBM Guardium*, transforma os dados consoante as configurações definidas, e carrega para os índices do *Elasticsearch*, este processo é explorado na secção 4.2.1.

Uma das funcionalidades mais poderosas do *Elasticsearch* é o sistema de agregações. Este permite calcular métricas e estatísticas (como contagens, médias, somatórios, etc.) diretamente sobre os dados indexados, sem necessidade de processamentos adicionais. Isto é fundamental para a construção de dashboards dinâmicos e análises exploratórias no *Kibana*.

O *Elastic Stack* é escalável graças à sua arquitetura distribuída, baseada na fragmentação de dados em *shards*⁹, replicação, balanceamento automático e execução paralela de consultas. Esta abordagem permite que o sistema mantenha a eficiência e uma latência reduzida, mesmo quando lida com grandes volumes de dados, adaptando-se facilmente através da adição de novos nós ao *cluster*¹⁰.

Por exemplo, num cenário em que se armazenam diariamente 10 milhões de registos de *logs*, o *Elasticsearch* pode dividir cada índice diário em 5 *shards*. Com um *cluster* composto por 3 nós, cada um desses nós armazenará e processará parte desses *shards*. Quando uma consulta é feita, ela é automaticamente distribuída por todos os nós, que processam os dados localmente e devolvem

⁸Interface que expõe recursos via HTTP usando regras REST (GET, POST, etc.).

⁹Partes de um índice distribuídas por vários nós para escalar desempenho.

¹⁰Grupo de nós que forma o sistema distribuído do *Elasticsearch*.

os resultados agregados ao nó coordenador. Se o volume de dados aumentar significativamente (por exemplo, para 30 milhões de registos diários), basta adicionar mais nós ao *cluster* para manter o desempenho, uma vez que os *shards* serão redistribuídos automaticamente e as consultas continuarão a ser executadas em paralelo.

Módulo de *machine learning* (ML)

O módulo de ML, utilizado para detetar *outliers*¹¹, baseia-se no comportamento dos utilizadores, que é criado através dos comandos SQL executados. O algoritmo é composto por duas componentes principais: a primeira avalia se as ações de um utilizador são consistentes com o seu histórico de comportamentos, enquanto a segunda analisa a consistência global, comparando as ações desse utilizador com as de outros utilizadores com perfis semelhantes. Cada anomalia detetada tem uma pontuação atribuída com base em fatores como a raridade do acesso, o volume, entre outros atributos. Os possíveis *outliers*, identificados com base num conjunto de acessos, são apresentados juntamente com uma pontuação calculada a partir das anomalias associadas.

2.2.4 HIDRA

O HIDRA é uma plataforma desenvolvida na MEO pela Direção de Cibersegurança e Privacidade (DCY). O HIDRA (*High Performance Infrastructure for Data Research and Analysis*) tem como características o seu alto desempenho e escalabilidade para realizar pesquisas distribuídas e análise de dados. O principal objetivo é apoiar as investigações de fraudes e de incidentes de cibersegurança. A plataforma é carregada com dados de diversas fontes utilizando processos de ETL (Extração, Transformação e Carregamento de dados) e baseia-se no *Elastic Stack*. Esta plataforma constitui a principal fonte de dados utilizada no âmbito do presente trabalho.

2.2.5 Docker

O Docker é uma plataforma que permite aos programadores desenvolver, distribuir e executar aplicações de forma consistente, independentemente do ambiente onde estas serão executadas. O *Docker* utiliza o conceito de *containers*, que são ambientes isolados capazes de reproduzir com exatidão o comportamento da aplicação definida numa imagem.

Imagem É um ficheiro de modelo (ou *template*) que define o conteúdo e comportamento de um *container*. Contém tudo o que é necessário para executar uma aplicação: código, bibliotecas, dependências e configurações do sistema. As imagens são criadas previamente e podem ser reutilizadas sempre que necessário.

Container É uma instância executável de uma imagem. Funciona como um ambiente isolado e leve que executa a aplicação conforme definido pela imagem de origem. As alterações realizadas durante a execução de um container (como instalação de pacotes ou modificações em ficheiros) não afetam a imagem base, permitindo iniciar sempre com um ambiente limpo e previsível.

¹¹ Observações que são significativamente diferentes quando comparadas com os restantes dados.

Para a construção das imagens utilizadas no projeto, foram criados ficheiros *Dockerfile*, que descrevem de forma declarativa todas as instruções necessárias para configurar o ambiente de execução. Nestes ficheiros estão definidas, por exemplo, as bibliotecas a instalar, a versão da linguagem de programação utilizada, os certificados digitais exigidos pelos serviços da organização, bem como scripts e configurações adicionais. Esta abordagem permitiu automatizar e uniformizar a criação de ambientes, garantindo que todos os membros da equipa, ou qualquer pessoa que precise de replicar o trabalho, consiga fazê-lo com facilidade e sem erros de configuração.

No contexto deste trabalho, o *Docker* foi utilizado para facilitar a configuração e execução dos diferentes componentes da aplicação desenvolvida. Com esta tecnologia, foi possível garantir que todos os serviços utilizados (como bases de dados e scripts de processamento) fossem executados de forma consistente e independente do sistema operativo utilizado. Isto permitiu reduzir problemas de compatibilidade e acelerar o processo de testes.

2.2.6 *Tableau*

A análise e interpretação eficazes de dados são fundamentais para a deteção de comportamentos anómalos em ambientes de base de dados. Neste contexto, o *Tableau* foi a ferramenta de visualização escolhida, dada a sua robustez, flexibilidade e capacidade de apresentar métricas de forma clara, interativa e acessível. Esta secção descreve o papel do *Tableau* na apresentação dos resultados da investigação, bem como a sua importância para a contextualização e compreensão dos casos de uso abordados ao longo do trabalho.

A escolha do *Tableau* justifica-se por diversos fatores. A ferramenta destaca-se pela sua capacidade de integração com uma grande variedade de fontes de dados, incluindo bases de dados relacionais, como MySQL, SQL Server e Oracle, bem como fontes não relacionais, como Elasticsearch. Além disso, o *Tableau* também suporta ficheiros em formatos como Excel e JSON. Neste caso específico, serão extraídos apenas dados provenientes das bases de dados relacionais e da fonte não relacional anteriormente referida. Adicionalmente, permite a criação de *dashboards* interativos que facilitam a filtragem dinâmica e a exploração detalhada das informações. Esta interatividade é crucial para uma análise exploratória eficaz, permitindo a rápida identificação de padrões, tendências e possíveis anomalias. Outro ponto relevante é o suporte a análises em tempo quase real, algo particularmente útil em contextos operacionais onde a resposta rápida a incidentes de segurança é crítica.

No âmbito deste projeto, o *Tableau* foi utilizado para representar visualmente os dados recolhidos, evidenciando métricas associadas à evolução do projeto. Através de representações gráficas intuitivas e hierarquizadas, foram criados *dashboards* que abordam diferentes dimensões da análise. Por exemplo, o caso de uso “*HouseKeeping* BD” apresenta uma visualização hierárquica de utilizadores detetados, segmentados por tecnologia, nome da base de dados, user e nível de severidade, complementada por uma análise temporal diária. A visualização criada para o caso de uso “Partilha de Contas” terá o mesmo objetivo, porém o resultado é diferente visto que a execução é baseada num espaço temporal, ou seja num certo intervalo pode ter mais casos do que noutro.

A utilização do *Tableau* não só reforça a argumentação desenvolvida ao longo do projeto como também demonstra, de forma prática, a aplicabilidade da *framework* proposta para detecção de anomalias. A capacidade de transformar dados técnicos complexos em representações compreensíveis e acionáveis contribui significativamente para a validação dos resultados obtidos e para o apoio à tomada de decisões em contextos de cibersegurança.

Capítulo 3

Trabalho Relacionado

Nesta secção pretende-se apresentar alguns projetos que acho serem relevantes para a realização deste projeto, tendo em conta as abordagens e metodologias utilizadas.

3.1 PoirotDB

O projeto de mestrado realizado por José Águas em 2020 tem como título “*Deteção de acessos anómalos a bases de dados críticas*” [3], e contém algumas bases e metodologias cruciais para o desenvolvimento deste projeto.

Nesse projeto, foi abordado o problema dos acessos anómalos a bases de dados, onde o objetivo desse projeto é diferente deste. O objetivo desse projeto era identificar acessos anómalos em bases de dados críticas da Altice Portugal, sob a monitorização da plataforma **IBM Guardium**. Para o efeito, foram utilizados dois módulos principais: o primeiro realiza a identificação de anomalias em quase tempo real através da implementação de políticas de segurança, enquanto o segundo identifica alterações no comportamento dos utilizadores com base em *outliers* detetados pelo algoritmo de *machine learning* (ML) do *IBM Guardium*.

Além de adotar uma metodologia diferente, os acessos a serem considerados são diferentes deste trabalho, estes incluem casos como origens incompatíveis com o tipo de conta, que verificam por exemplo se uma conta realiza um acesso através de um IP que não é comum, exfiltração de informação, que verifica se uma conta acede a mais de cinquenta registos de uma base de dados em menos de um minuto, entre outros.

3.2 PoirotDBv2

No ano seguinte (2021), foi realizado um projeto por Ivo Pascoal, tendo como base o projeto referido anteriormente, com o título “*PoirotDBv2: Deteção, qualificação, avaliação e resposta rápida a incidentes de acesso ilícito ou exfiltração ilícita de informação sensível*” [17].

Nesse projeto, o problema abordado são os acessos ilícitos ou exfiltração ilícita de informação sensível, uma continuação do projeto **PoirotDB**, no entanto, o objetivo era diferente. O objetivo desse projeto era o estudo do funcionamento e das capacidades do módulo de ML do *IBM Guard-*

dium para detetar, qualificar, avaliar e responder rapidamente a incidentes de acesso ilícito ou exfiltração ilícita de informação sensível.

Além disso, no caso de serem detetados acontecimentos ilícitos, pretendeu-se ainda que o número de falsos positivos¹ e de falsos negativos² diminuísse, e que houvesse uma correlação entre os eventos de bases de dados e eventos das estações de trabalho.

Assim foi implementado o uso do *IBM QRadar UBA (User Behavior Analytics)*, ferramenta disponível no *IBM QRadar*³. Esta tem a capacidade de, através de regras, filtrar os vários dados e com base nos modelos de ML aprender o comportamento de um dado utilizador, e perceber em que alturas é que o mesmo está a caminhar num sentido inverso ao habitual.

O principal objetivo do uso desta ferramenta é a possibilidade de extração de eventos como horas de autenticação na estação de trabalho, websites visitados, introdução de *pen drives*, o que não está disponível usando apenas o *IBM Guardium*. Um exemplo seria a deteção de exfiltração de muitos dados por parte do *IBM Guardium*, depois no *IBM QRadar UBA* verifica-se que houve uma introdução de ficheiros em flash drives numa estação de trabalho cujo IP coincide entre os dois eventos, conseguindo detetar uma incidente grave.

Assim é possível a correlação dos dados entre os eventos de bases de dados e eventos das estações de trabalho, conseguindo uma análise mais detalhada e uma avaliação mais precisa dos acontecimentos.

O projeto também aborda o tema da partilha de contas, identificado como um dos comportamentos ilícitos ao longo do seu desenvolvimento. Neste contexto, é apresentada uma análise teórica inicial, juntamente com a identificação de dois tipos de comportamentos relacionados com a partilha de contas:

- Quando existem vários utilizadores a acederem à mesma conta de base de dados;
- Um utilizador usar uma conta de base de dados que não é dele.

Na secção deste projeto onde exploro o mesmo tema, além de otimizar e melhorar alguns processos, tem-se como objetivo ampliar a zona de deteção com mais situações encontradas:

- Utilizador utiliza mais do que um computador;
- Utilizador utiliza apenas um computador mas verifica-se que não é dele.

¹Resultados incorretamente identificados como verdadeiros.

²Resultados incorretamente identificados como falsos.

³<https://www.ibm.com/docs/en/qradar-common?topic=app-qradar-user-behavior-analytics>

Característica	PoirotDB	PoirotDB v2	Presente Trabalho
Objetivo	Deteção de acessos anómalos em bases de dados críticas usando apenas o <i>IBM Guardium</i>	Estudo do módulo de ML do <i>IBM Guardium</i> com correlação de eventos externos	Deteção de acessos suspeitos de utilizadores nominais em BDs usando o <i>IBM Guardium</i> e o <i>Elasticsearch</i>
Ferramentas	<i>IBM Guardium</i> (Políticas + Grupos, Módulo ML)	<i>IBM Guardium</i> (Módulo ML) + <i>IBM Qradar UBA</i>	<i>IBM Guardium</i> (Políticas + Grupos) + <i>Elasticsearch</i>
Casos de uso	Exfiltração de dados, <i>brute force</i> , <i>SQL injection</i>	Incidentes de acesso ilícito validados com eventos de estação de trabalho	<i>HouseKeeping</i> BDs, Partilha de Contas, Monitorização DBAs
Metodologia	Baseada em regras e comportamento do <i>IBM Guardium</i>	Correlação de múltiplas fontes (<i>IBM Guardium</i> + <i>Qradar</i>)	Baseada em regras e políticas de segurança do <i>IBM Guardium</i>
Diferencial	Foco exclusivo nas funcionalidades nativas do <i>IBM Guardium</i> e nos dados de BD	Integração entre eventos de BD e eventos de estações de trabalho para validação cruzada e redução de falsos positivos	Integração de <i>IBM Guardium</i> e <i>Elasticsearch</i> com deteção por regras focada em utilizadores nominais e padrões que escapam às soluções tradicionais.

Tabela 3.1: Comparação entre os projetos PoirotDB, PoirotDB v2 e o presente trabalho.

3.3 TabSec

O projeto *TabSec: A Collaborative Framework for Novel Insider Threat Detection* [9] propõe uma arquitetura colaborativa que combina sistemas de deteção de intrusões (IDS) com análise comportamental de utilizadores e entidades (UEBA), a fim de detetar ameaças internas com maior precisão. Esta abordagem procura superar limitações das soluções convencionais, como o *IBM Guardium*, especialmente em cenários onde ataques externos evoluem para acessos internos, por exemplo, através de ataques do tipo *User to Root* (U2R) [4] e *Remote to Local* (R2L) [5].

A principal inovação do TabSec está na utilização do modelo TabNet, que aplica mecanismos de atenção seletiva para identificar automaticamente as características mais relevantes dos dados em cada decisão. Isto permite lidar eficazmente com dados desbalanceados e detetar ataques de classe rara, um problema típico em ambientes reais de segurança. O modelo foi avaliado com os conjuntos de dados NSL-UEBA e KDD-UEBA, atingindo uma precisão média entre 96,7% e 97,3%, superando técnicas tradicionais como XGBoost, CatBoost e LGBM. Além disso, mostrou melhor desempenho em termos de taxa de falsos positivos, deteção de ataques mascarados e estabilidade durante o treino.

⁴Ataque em que um utilizador comum obtém acesso de administrador.

⁵Ataque onde um atacante externo obtém acesso local ao sistema.

Tabela 3.2: Comparação entre os projetos PoirotDB, PoirotDBv2 e TabSec.

Aspeto	PoirotDB / PoirotDBv2	TabSec
Tecnologia principal	<i>IBM Guardium</i> (com <i>QRadar UBA</i> no PoirotDBv2)	UEBA + IDS integrados com TabNet
Tipo de <i>Machine Learning</i>	Deteção baseada em outliers e regras definidas	<i>Deep learning</i> com atenção seletiva (TabNet)
Adaptabilidade	Regras pré-definidas com thresholds estáticos	Seleção dinâmica de atributos relevantes por instância
Cobertura de ataques	IPs invulgares, acessos rápidos, partilha de contas	Ameaças internas e externas, incluindo U2R e R2L
Correlação de eventos	Eventos em bases de dados e estações de trabalho (PoirotDBv2)	Correlação entre diferentes domínios de ataque (IDS + UEBA)
Precisão / Estabilidade	Não quantificada (análise descritiva)	96,7% a 97,3% de precisão média, baixa taxa de falsos positivos
Limitação notável	Dependência de regras específicas e ferramentas IBM	Requer afinação cuidadosa de hiperparâmetros

3.4 PID(iscover)

O projeto de mestrado realizado por Raúl Koch em 2022, intitulado "*Descoberta e classificação automática de informação crítica em bases de dados*", propõe uma metodologia para a deteção sistemática de bases de dados com informação crítica em redes empresariais de larga escala. Como ferramenta, foi utilizado o *IBM Guardium*, em particular a funcionalidade de **descoberta e classificação de dados** da solução *Database Activity Monitoring* (DAM), cujo objetivo é apoiar o armazenamento, gestão e segurança de tipos específicos de dados [13].

A metodologia centra-se na identificação de bases de dados que utilizem tecnologias como *Oracle*, *Informix*, *MySQL* e *SQL Server*, seguida da aplicação da funcionalidade *Auto-discovery* do *IBM Guardium* para localizar e classificar informação crítica. O processo de deteção envolve inicialmente a análise do formato, estrutura e contexto dos dados, bem como a definição do que deve ser considerado crítico e respetiva prioridade, culminando na criação de regras específicas para essa classificação.

O projeto resultou na criação de um conjunto de dados classificados como críticos, que são mantidos atualizados através de varrimentos periódicos.

Capítulo 4

Fontes de Dados e Implementação nas Ferramentas

Neste capítulo são descritas as fontes de dados utilizadas no projeto, os processos de preparação e tratamento desses dados, bem como o trabalho prático realizado nas ferramentas IBM *Guardium* e *Elasticsearch*. O objetivo é fornecer uma visão completa do ambiente técnico necessário para a definição e implementação dos casos de uso apresentados nos capítulos seguintes.

4.1 Fontes de Dados

Para a realização deste trabalho, foi essencial identificar as fontes de dados relevantes, compreender a natureza da informação nelas contida e definir como utilizá-la de maneira eficiente. Esta subsecção apresenta uma visão geral das diversas bases de dados utilizadas, explicando as suas funções, os tipos de informações que armazenam e a sua relevância para o estudo.

- A base de dados AD refere-se ao *Active Directory*, uma solução da *Microsoft* amplamente utilizada para gerir identidades e recursos em redes corporativas. O *Active Directory* é um serviço de diretório que armazena e organiza informações relacionadas a utilizadores, computadores, grupos e políticas da organização;
- A base de dados GI é mantida pela ferramenta responsável por gerir o ciclo de vida das identidades dentro da organização. Esta automatiza processos como criação, alteração e eliminação de contas;
- A base de dados dos recursos humanos (RH) armazena informações relacionadas aos colaboradores da organização.
- A base de dados EXC contém informações sobre exceções aplicadas às contas dos colaboradores. Um exemplo de exceção é uma baixa médica, onde o utilizador ficará inativo durante o período da baixa, e a exceção servirá para impedir a eliminação do mesmo até voltar ao ativo;

- A base de dados denominada como “ETLS” armazena informações sobre os utilizadores ativos em bases de dados, bem como o estado e permissões dos mesmos, nome e tecnologia da base de dados, e se o mesmo se encontra na ferramenta GI;
- Dados provenientes dos logs das bases de dados monitorizadas pelo *IBM Guardium*, armazenados no *Elasticsearch*. Estes registos incluem informações sobre erros, *logins* e *logins* falhados, ações executadas e outros eventos relevantes.
- **Grupo de dados críticos** definido no *IBM Guardium*, contendo etiquetas que identificam informação sensível, como números de telemóvel, moradas e outros dados críticos.

As bases de dados RH, AD e GI são as principais fontes de informação relacionadas às contas dos colaboradores, fornecendo dados como o nome do colaborador, nome de utilizador, número de colaborador, estado atual na empresa, entre outros campos essenciais para a caracterização dos utilizadores e produção de evidências em casos suspeitos. Na **tabela 1** é apresentado um exemplo de correlação das várias fontes de dados, onde é mostrado um utilizador que se encontra com o estado “Ativo” em todas as fontes de dados.

Tabela 4.1: Exemplo de correlação de dados.

Nome do colaborador	Número de colaborador	Estado AD	Estado GI	Estado EXC	Estado RH
Francisco António	123456789	Ativo	Ativo	Ativo	Ativo

As fontes de dados RH e GI disponibilizam datas que servem de referência para a saída de um colaborador. No entanto, a AD não contém esta informação. Por este motivo, recorreu-se a um índice do *Elasticsearch* que armazena logs de eventos gerados pelo *Windows*. Esses eventos são identificados por um *ID*, sendo que o evento de desativação de conta corresponde ao ID 4725 [16]. Assim, através de uma *query* ao *Elasticsearch*, foi possível, a partir do utilizador e do ID do evento, extrair a data de desativação associada a esse utilizador.

4.2 Processamento dos dados

O tratamento e análise de dados extraídos para utilização em *scripts* de processamento requerem, frequentemente, uma etapa de transformação prévia. Embora os dados já estejam armazenados no *Elasticsearch*, cada fonte tem a sua própria forma de estruturar e indexar a informação. Para juntá-los numa análise única, é preciso antes uniformizar esses formatos. Por exemplo, ao comparar utilizadores oriundos de diferentes fontes, é comum que surjam variações como letras maiúsculas e minúsculas, presença ou ausência de domínios, e outras inconsistências de formatação. Estas divergências dificultam o manuseamento automatizado e exigem um trabalho de uniformização dos dados logo após a sua extração.

Neste contexto, o processamento dos dados assume um papel crucial, ao permitir aplicar transformações que tornem os dados mais limpos, coesos e semanticamente comparáveis. Esta etapa, apesar de suceder à extração propriamente dita, funciona como um pré-tratamento necessário antes de análises mais aprofundadas, agregações, ou visualizações. Na subseção seguinte, detalha-se a abordagem adotada para normalizar os dados de forma sistemática, corrigindo limitações identificadas em processos anteriores.

4.2.1 Normalização dos dados

Anteriormente, identificou-se um problema na extração de dados a partir do *Elasticsearch*, os dados não se encontravam devidamente normalizados, o que obrigava o *script* a realizar normalizações *a posteriori*, atuando apenas sobre os dados extraídos e não sobre os eventos em si. Esta abordagem mostrava-se ineficiente, pois obrigava todos os processos que consumissem esses dados a repetir o mesmo tratamento. Ao realizar a normalização já na fase de transformação do processo de ETL, garante-se que os dados ficam uniformizados na origem, simplificando e padronizando o consumo pelas diferentes aplicações.

Para resolver esta incongruência, o processo ETL foi revisto e ajustado, passando a incluir, na fase de transformação, a normalização dos dados de forma sistemática. Adicionalmente, os dados passaram a ser estruturados em campos mais significativos e semanticamente relevantes, facilitando a sua posterior análise, indexação e visualização. Esta melhoria permitiu garantir maior consistência nos dados armazenados no *Elasticsearch* e assegurou que o processo de extração se limitasse à sua função principal, sem sobrecarregar essa fase com tarefas que pertencem a etapas anteriores do *pipeline*. Desta forma, é igualmente possível compreender de que modo os dados extraídos do *IBM Guardium* são posteriormente extraídos, transformados e carregados no *Elasticsearch*. A Figura 4.1 apresenta o processo de ETL correspondente, estabelecendo continuidade com as figuras 4.3 e 2.6.

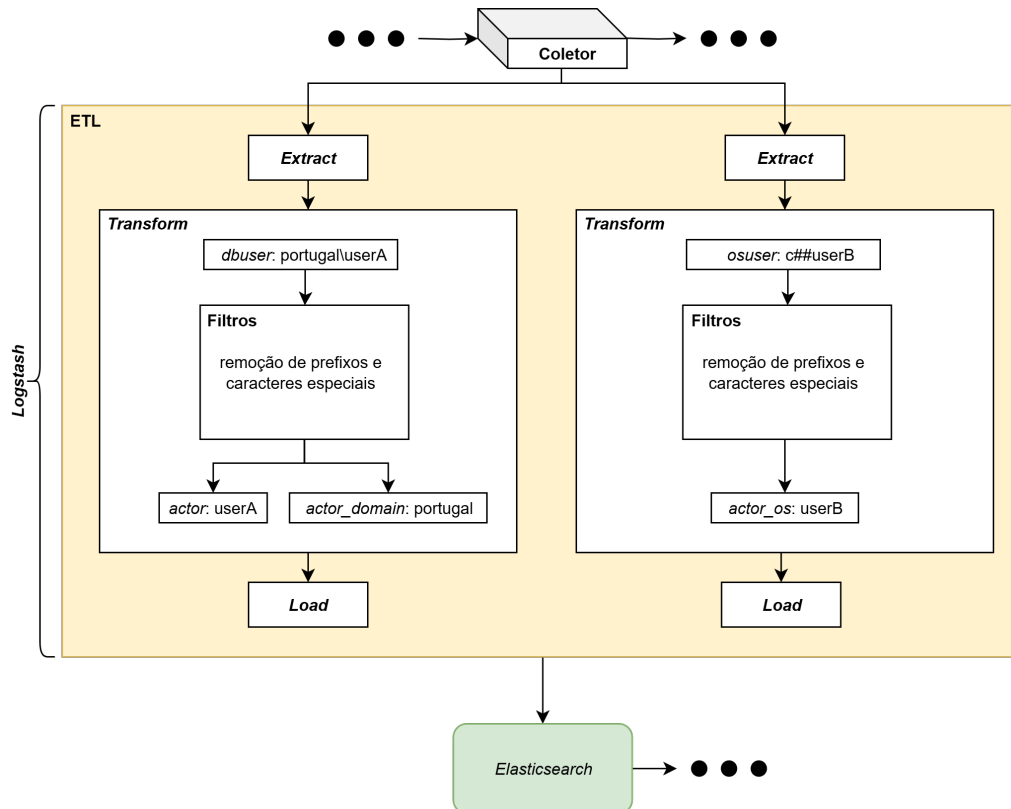


Figura 4.1: Processo de ETL do *IBM Guardium* para o *Elasticsearch*.

4.2.2 Filtragem dos dados

Durante o processo de extração de dados a partir do *Elasticsearch*, verificou-se que, para garantir histórico suficiente para a análise, era necessário consultar um período de 30 dias. No entanto, este intervalo resultava na obtenção de cerca de 120 milhões de registos numa única consulta, o que comprometia o desempenho do sistema e tornava a análise subsequente demorada e complexa.

A redução do período temporal, embora pudesse aliviar o volume de dados, não foi considerada solução viável, pois implicaria risco de perda de casos relevantes. Por exemplo, uma partilha de contas com um intervalo de 30 dias entre eventos deixaria de ser detetada caso a janela fosse reduzida para 20 dias. Assim, optou-se por manter a janela de 30 dias como equilíbrio entre cobertura e eficiência, embora, idealmente, a análise se estendesse a um ano para capturar todos os casos possíveis.

Para mitigar o problema do volume excessivo, o processo de extração foi revisto e dividido em etapas complementares:

- **Filtragem por tipo de evento:** Identificou-se que uma parte significativa dos registos correspondia a eventos de baixo valor para a análise, como erros genéricos de SQL, *logins* falhados ou operações irrelevantes para o objetivo do estudo. Foram então introduzidos filtros específicos na query SQL, restringindo os resultados apenas a eventos pertinentes, como *logins* bem-sucedidos.

- **Seleção de campos essenciais:** As tabelas originais possuíam dezenas de campos por registo, muitos sem utilidade para o processamento subsequente. Ao extrair apenas os atributos estritamente necessários, reduziu-se de forma significativa o volume de dados transferidos e processados, aumentando a eficiência global.

Com estas otimizações, obteve-se uma redução substancial no volume de dados processados, preservando a informação essencial para a análise e garantindo ganhos significativos de desempenho tanto na fase de extração como no processamento posterior no ETL. Segue abaixo um exemplo de *query* já com a aplicação dos filtros descritos anteriormente.

Consulta SQL para obter utilizadores ativos no Elasticsearch

```
SELECT "user"
FROM indice_elasticsearch
WHERE ts >= '{data_inicio}' AND ts <= '{data_fim}'
AND user NOT IN ('?', '', '_', 'None', 'Null') AND actions = 'Login'
GROUP BY "user";
```

4.3 IBM Guardium

Nesta secção pretende-se mostrar como foi feita a criação de grupos dinâmicos e a implementação de políticas na ferramenta *IBM Guardium* para os casos de uso.

4.3.1 Criação do grupo

Para o desenvolvimento do caso de uso de Monitorização de DBAs, é necessário criar um grupo que permita identificar os DBAs. Nesse sentido, foram definidas as etapas necessárias para a criação do grupo, ilustradas na Figura 2.2.

Após a definição do grupo, tornou-se essencial dispor de uma fonte de dados que reunisse todos os DBAs, possibilitando assim a execução de uma instrução *SELECT* para a sua identificação.

Com a criação dessa fonte de dados, o fluxo resultante encontra-se representado na Figura 4.3, constituindo uma extensão do processo mostrado na Figura 2.2.

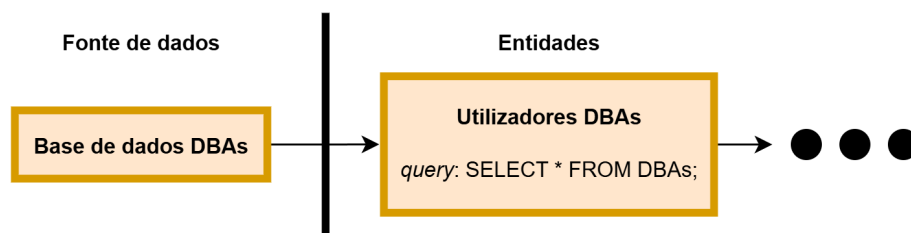


Figura 4.2: Criação da fonte de dados integrada no processo de criação do grupo de DBAs.

4.3.2 Criação da política

No *IBM Guardium*, as políticas constituem o mecanismo central para a definição e aplicação de regras de monitorização, auditoria e geração de alertas sobre atividades realizadas em bases de

dados. Essas políticas desempenham um papel essencial na detecção de comportamentos anómalos, na prevenção de violações de segurança e na garantia de conformidade com regulamentações, como o RGPD.

A Figura 4.3 apresenta o processo de criação de uma política direcionada para a monitorização de acessos a dados críticos realizados por DBAs, servindo como referência para a implementação prática desta abordagem.

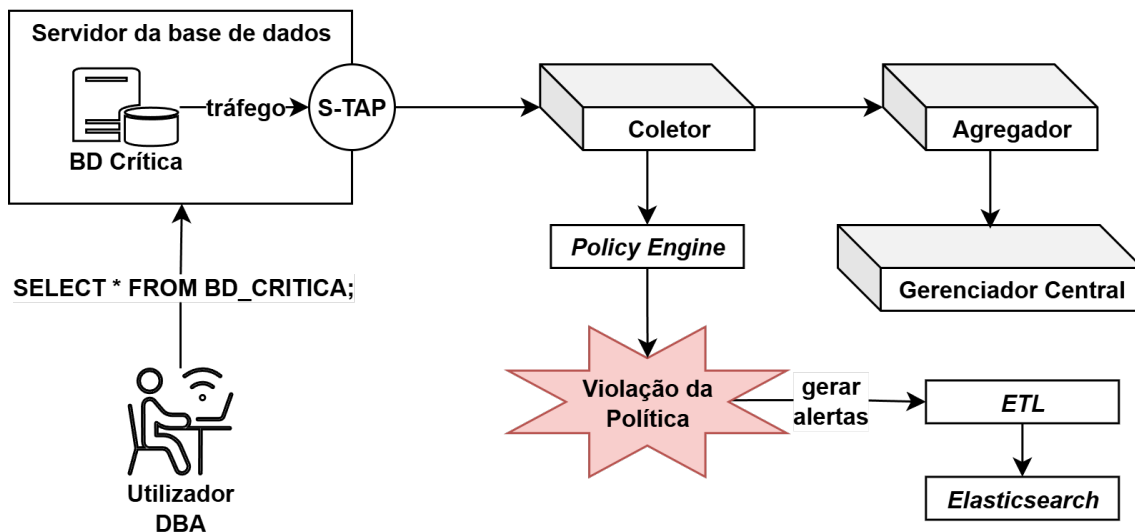


Figura 4.3: Criação da política de monitorização de acessos a dados críticos por DBAs (adaptado de [17]).

4.4 Elasticsearch

Nesta secção analisam-se as diferentes formas de interação com o *Elasticsearch* exploradas ao longo do projeto, com especial foco na eficiência das consultas e no impacto do método escolhido sobre o desempenho.

4.4.1 Abordagens Disponíveis

Durante o desenvolvimento, foram estudadas diversas formas de obter dados do *Elasticsearch*:

- **Consulta via API REST com DSL:** método nativo do *Elasticsearch*, em que as consultas são escritas directamente na *Domain Specific Language* (DSL). Oferece controlo total sobre a consulta, mas exige uma construção manual detalhada e pode ser menos eficiente em grandes volumes de dados devido à ausência de optimizações automáticas.
- **Consulta via *Elasticsearch SQL*:** permite escrever consultas em SQL, que são traduzidas internamente para DSL. Esta camada aplica optimizações automáticas, como *filter push*-

down^[1] e *aggregation pruning*^[2], podendo ser mais eficiente do que a utilização directa de DSL.

- **Exportação via *curl* ou *wget*:** execução de consultas directamente pela linha de comandos, retornando dados em formato JSON para processamento externo.
- **Logstash com output em ficheiro:** configuração do *Logstash* para exportar dados para ficheiros JSON, que podem ser lidos posteriormente por scripts.

4.4.2 Discussão das abordagens

Entre todas as alternativas, destacam-se a API REST com DSL e o *Elasticsearch SQL* como as mais adequadas para o fluxo de trabalho deste projeto, visto que as restantes não eram fiáveis para a exportação de grandes volumes de dados. A API REST oferece controlo total e flexibilidade, enquanto o SQL traz simplicidade e optimizações automáticas que reduzem a carga de processamento. Apesar de ser traduzido para DSL internamente, o SQL consegue, em muitos casos, superar o desempenho da escrita manual em DSL, especialmente se esta não estiver optimizada.

4.4.3 Comparação de Desempenho: API vs SQL

Para validar qual das duas abordagens seria mais vantajosa, realizaram-se testes utilizando a mesma máquina e mantendo fixo o intervalo temporal e, consequentemente, o volume de dados processado. Cada teste foi executado 5 vezes e os valores apresentados correspondem à média dos tempos de execução. Antes de cada medição, realizou-se um *dry-run*^[3], de forma a garantir que as medições refletem de forma mais fiel o desempenho real das consultas.

Tabela 4.2: Comparação dos testes de desempenho entre consulta via API e consulta via SQL.

Janela temporal (h)	Volume de dados	Tempo execução API (s)	Tempo execução SQL (s)	Ganho de tempo entre SQL e API (s)
1	189 175	35.13	9.05	26.08
6	1 151 922	284.34	41.38	242.96
12	1 944 501	491.24	71.97	419.27
24	3 580 991	804.24	124.12	680.12

Como se observa, a camada SQL apresentou tempos de resposta significativamente inferiores, mesmo em volumes mais pequenos, mantendo eficiência consistente em grandes volumes de da-

¹Aplicação antecipada de filtros para reduzir dados processados

²Eliminação de dados irrelevantes antes da agregação

³Execução preliminar da *query* sem registo dos resultados, com o objetivo de preparar a cache do *Elasticsearch* e reduzir o impacto de leituras iniciais de disco. Esta abordagem permite que as medições subsequentes dos tempos de execução reflitam de forma mais fidedigna o desempenho real das consultas.

dos. Isto confirma que o *Elasticsearch SQL* é mais vantajoso em termos de desempenho, enquanto a API REST continua indicada para cenários de consultas rápidas e simples.

4.4.4 Considerações Finais

O equilíbrio entre desempenho, simplicidade e flexibilidade sugere que a utilização de API REST é mais adequada em cenários de consultas rápidas, enquanto o *Elasticsearch SQL* se destaca em contextos mais complexos. Dado que o cenário em análise apresenta elevada complexidade, a opção recai sobre o uso de *Elasticsearch SQL*. Em ambos os casos, a aplicação de boas práticas de modelação de dados e optimização de queries é fundamental para assegurar um desempenho consistente.

Capítulo 5

Casos de Uso

Neste capítulo são apresentados os casos de uso desenvolvidos, abordando primeiramente o processo de definição e, posteriormente, os detalhes da sua implementação.

5.1 Definição

Esta secção tem como objetivo apresentar e contextualizar os casos de uso definidos. Todos os casos de uso foram implementados em *Python* e em *container* com *Docker*, assegurando execução automática e reproduzível. Estes casos de uso foram definidos com o objetivo de detetar acessos suspeitos. Apesar dos casos de uso cobrirem cenários diferentes, acabam por ter o mesmo objetivo depois de detetarem acessos suspeitos: gerar evidências, efetuar *reporting* automático, e gerar métricas, como é possível verificar na Figura 5.1

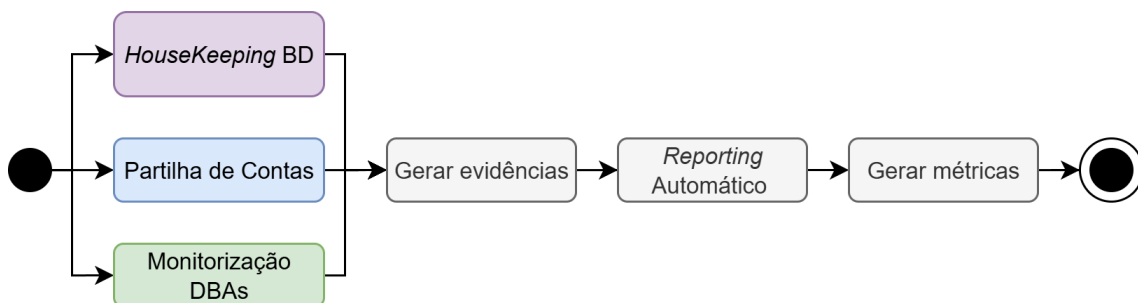


Figura 5.1: Arquitetura macro geral.

5.1.1 *HouseKeeping* BD

Um dos aspetos fundamentais na gestão eficaz de sistemas e processos corporativos é a realização periódica de tarefas de *housekeeping* (HK). Esse termo, originalmente associado à organização doméstica, no contexto empresarial refere-se à manutenção regular e à limpeza de dados, processos e sistemas para garantir a sua integridade, segurança e eficiência.

No âmbito das contas de utilizadores, o HK assume uma importância crítica, pois envolve identificar e lidar com contas nominais de colaboradores que já não fazem parte da organização. Essas

contas, se não forem devidamente geridas, podem representar riscos significativos à segurança. Entre os principais riscos estão os acessos não autorizados, nomeadamente quando contas antigas são utilizadas por terceiros ou continuam ativas em processos automatizados após a saída do colaborador. Além disso, existe o risco de elevação de privilégios, por exemplo, se uma conta com permissões elevadas for reutilizada indevidamente para obter acesso privilegiado a sistemas ou dados sensíveis, contornando os mecanismos normais de controlo de acessos.

Importa ainda distinguir entre os diferentes tipos de contas de base de dados no que diz respeito ao seu mecanismo de autenticação. Algumas contas de BD estão configuradas para autenticar diretamente com a *Active Directory* (AD) da organização. Nestes casos, os processos regulares de HK da AD asseguram que, quando um colaborador sai da empresa, a sua conta na AD é automaticamente desativada ou removida. Como consequência, a conta de BD associada torna-se inutilizável, uma vez que a autenticação deixa de ser possível, o que mitiga significativamente os riscos associados à permanência da conta na BD.

Por outro lado, existem contas de BD que não dependem da AD para autenticação, funcionando como contas locais no sistema de base de dados. Nestes casos, a saída do colaborador não aciona automaticamente qualquer processo de HK, existindo a probabilidade destas contas permanecerem criadas na base de dados, mesmo após a saída do utilizador da organização.

Neste contexto, o presente caso de uso tem como objetivo identificar e corrigir contas de bases de dados que permanecem existentes apesar de os colaboradores associados já não fazerem parte da organização. O processo inicia-se com a recolha e análise de informação proveniente de diversas fontes de dados, sendo esta análise utilizada para identificar casos que devem ser encaminhados para o processo de HK. Caso não sejam encontrados casos, o processo termina; caso contrário, são geradas evidências que comprovam que a conta de base de dados continua ativa e que o colaborador correspondente já deixou a empresa. Em seguida, verifica-se a existência de acessos realizados após a data de saída do colaborador. Caso existam, os logs dessas ações são adicionados às evidências previamente geradas e registados numa tabela de base de dados. Por fim, é enviado um email contendo todas as evidências necessárias para a resolução da situação.

A arquitetura macro geral do processo de HK aplicado às bases de dados é apresentada na Figura 5.2.

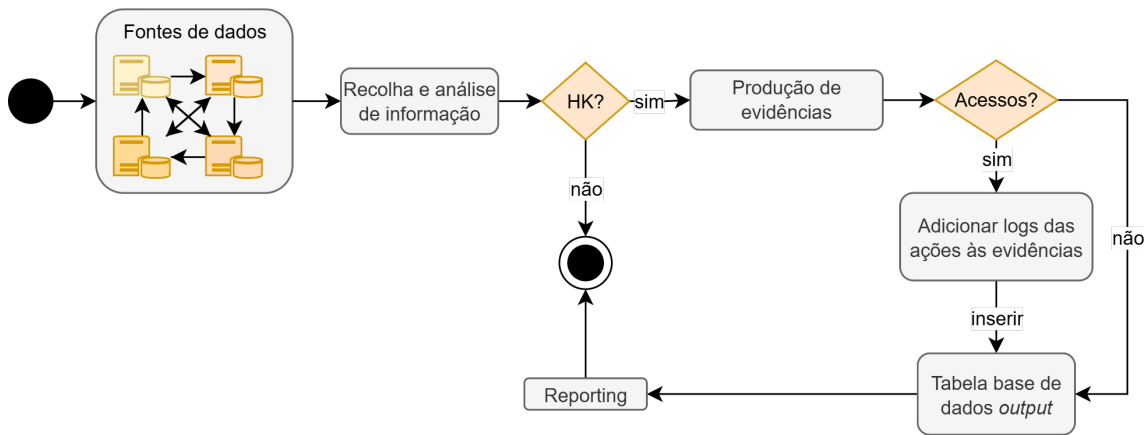


Figura 5.2: Arquitetura macro geral de *HouseKeeping* BD.

5.1.2 Partilha de Contas

O caso de uso “Partilha de Contas” tem como objetivo identificar situações de partilha de contas no contexto da gestão de acessos e da conformidade. Tal prática é considerada uma não conformidade grave, uma vez que pode comprometer a rastreabilidade, a segurança e a integridade dos sistemas de informação. Em conformidade com as boas práticas de segurança, cada utilizador deve dispor, no máximo, de uma conta nominal associada. Além disso, poderá ainda ter acesso a contas aplicacionais, desde as mesmas sejam usadas por sistemas e/ou integrações automáticas.

A utilização inadequada dessas contas, nomeadamente através do uso de credenciais roubadas pertencentes a terceiros ou da partilha de uma mesma conta por múltiplos utilizadores deve ser detetada e analisada. A partilha de credenciais não só dificulta a identificação de ações individuais no sistema, como também representa um risco significativo para a segurança organizacional.

Para apoiar este processo, foi concebida uma arquitetura macro que descreve, de forma geral, o mecanismo de identificação e análise de situações de partilha de contas. Essa visão de alto nível encontra-se representada na Figura [5.3](#)

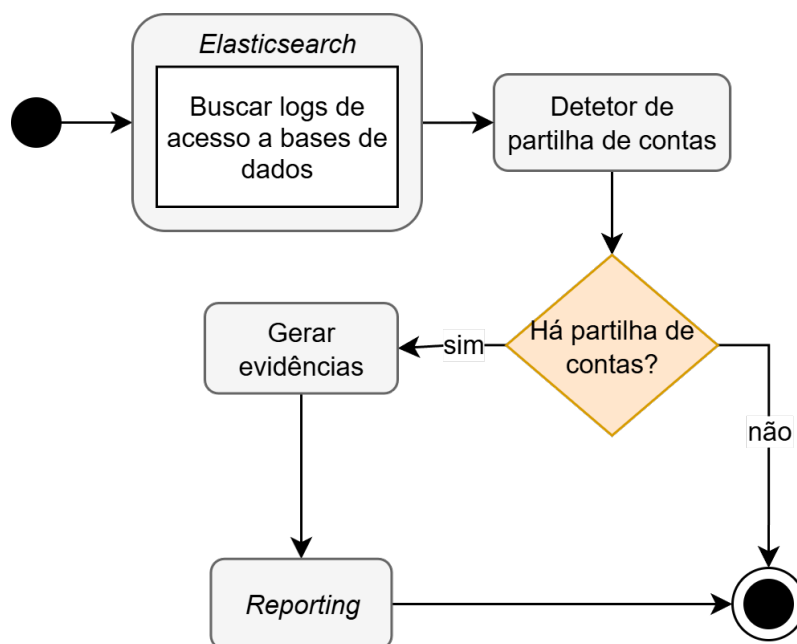


Figura 5.3: Arquitetura macro geral de Partilha de Contas.

5.1.3 Monitorização DBAs

Os administradores de bases de dados (DBAs) detêm acessos privilegiados a sistemas que armazenam e processam informações sensíveis da organização. Embora o seu papel seja fundamental para a operação e manutenção desses sistemas, os privilégios de que dispõem representam riscos significativos, sobretudo se forem utilizados para aceder ou manipular dados sem a devida autorização.

Neste caso de uso, é projetado um processo automatizado de execução periódica responsável por detetar acessos a dados críticos por parte de DBAs. O seu objetivo é identificar todas as contas de DBAs existentes na organização e armazená-las numa base de dados especificamente criada para esse fim. Posteriormente, será configurado um grupo na ferramenta *IBM Guardium*, que consultará essa base de dados e incluirá automaticamente as contas identificadas. Este grupo será associado a uma política de monitorização que permite verificar se algum dos seus membros acedeu a dados sensíveis previamente definidos noutro grupo já existente na ferramenta.

De forma a ilustrar este processo, a Figura 5.4 apresenta a arquitetura macro concebida para suportar a monitorização de DBAs, evidenciando os principais componentes e fluxos envolvidos.

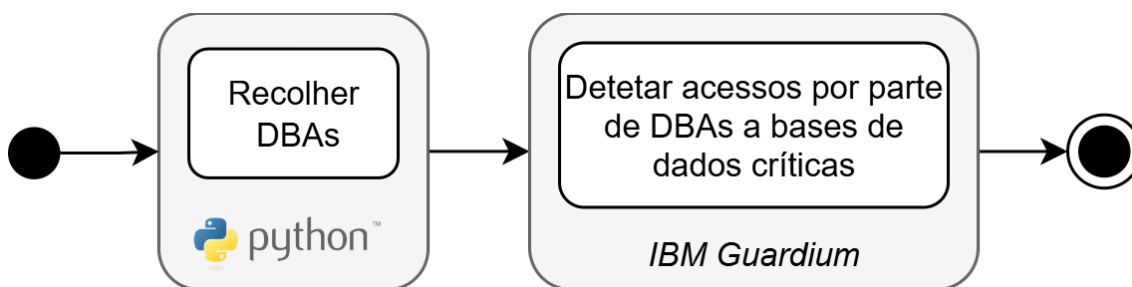


Figura 5.4: Arquitetura Macro DBAs.

5.2 Implementação

Neste capítulo apresenta-se o processo micro de implementação de cada caso de uso, incluindo a descrição das etapas seguidas, os testes realizados, bem como os resultados e conclusões obtidas.

5.2.1 *HouseKeeping* BD

O processo de *HouseKeeping* BD está estruturado em quatro subprocessos principais, cada um com objetivos e métodos específicos, o que permite uma maior clareza, controlo e flexibilidade no processo.

Deteção de Necessidade de *housekeeping*

O primeiro subprocesso consiste em identificar contas que estejam envolvidas em processos ativos, mas que possam necessitar de ações de HK. Para isso, inicia-se uma consulta a diversas fontes de dados, incluindo as informações provenientes do departamento de Recursos Humanos (RH), os dados da plataforma de gestão de identidades da MEO (GI) e os registos do *Active Directory* (AD).

Após a recolha, os dados são submetidos a um processo de normalização, no qual são eliminadas entradas com valores nulos, vazios ou inválidos. Com os dados limpos, são avaliados determinados critérios para determinar se uma intervenção de HK é necessária. São considerados casos que indicam necessidade de intervenção, por exemplo, quando se verifica que uma conta de um colaborador que já saiu da organização, se encontra ainda presente numa BD. Por outro lado, não se procede a ações de HK em situações em que o RH, o AD ou a plataforma GI indicam um estado ativo da conta, ou quando existe uma exceção ativa.

Caso não sejam encontradas contas que necessitem de intervenção, este subprocesso é concluído. Em contrapartida, quando identificadas, as informações são consolidadas num relatório que é armazenado numa tabela de base de dados de *output* do projeto.

Verificação de Acessos

No segundo subprocesso, o foco recai sobre a análise dos acessos das contas identificadas previamente, para verificar se estas continuam a ser utilizadas após a data de saída dos colaboradores da organização. Inicialmente, são lidos os registos da tabela de BD de *output* referida anteriormente, que contém os utilizadores identificados para HK. Posteriormente, procede-se à análise de um índice, que contém os *logs* de acessos a bases de dados, no *Elasticsearch*, com o objetivo de identificar possíveis acessos indevidos realizados após a saída de cada utilizador. Para tal, é utilizada uma *query* (anonimizada no exemplo abaixo) que segue a lógica do processo: para cada utilizador identificado no contexto do *housekeeping* (designado como `userHK`), verifica-se se existem registos de acesso posteriores à sua data de saída (`data_saida_userHK`).

Caso sejam encontrados acessos nesse intervalo, a consulta devolve campos tais como a máquina utilizada, nome da base de dados e datas de acessos.

Consulta SQL para obter utilizadores ativos no Elasticsearch

```
SELECT "user_device", "bdName", "timestamp"
FROM "elasticsearch_index"
WHERE "actor" = 'userHK' and ts >= 'data_saida_userHK'
```

As contas que apresentarem acessos ativos depois da saída são registadas numa nova tabela de *output* que irá conter as contas de base de dados que continuam a ter acessos. A partir deste ponto, realiza-se uma junção entre esta tabela e a tabela de *output* do processo, originando uma tabela final. Esta última tabela consolida tanto as evidências da necessidade de *housekeeping* quanto os eventos de acesso identificados.

Cálculo da Severidade

Apesar de todas as não conformidades terem de ser corrigidas, é calculada a severidade para priorizar a remediação de contas em incumprimento. Este índice reflete a gravidade de um incidente associado a cada conta, considerando fatores como o estado da conta, o tipo de utilizador e a existência de acessos após a data de saída.

Os fatores considerados são apresentados na Tabela 5.1. A atribuição dos valores baseia-se na gravidade da severidade associada a cada condição:

- **estadoContaScore:** Contas ativas apresentam maior severidade, pois permanecem operacionais e podem ser utilizadas imediatamente de forma indevida, recebendo assim valor 0.5. Já as contas bloqueadas ou expiradas apresentam menor severidade, mas ainda são relevantes, uma vez que podem ser reativadas ou podem ser identificadas em auditorias resultando em não conformidades, recebendo valor 0.3. Este valor reflete que a severidade não desaparece completamente mesmo quando a conta não está ativa.
- **tipoContaScore:** Utilizadores com privilégios de administração apresentam maior severidade, pois o tipo de ações possíveis de ser executadas podem ter um maior impacto, rece-

bendo valor 0.5. Utilizadores sem privilégios administrativos apresentam menor severidade e, por isso, recebem valor 0.1.

- **acessosScore:** A existência de acessos após a saída aumenta a severidade, indicando contas que continuam operacionais indevidamente, atribuindo-lhes valor 1. A ausência de acessos após a saída reduz a severidade relativa, refletindo menor impacto, e por essa razão recebe valor 1.5. Este fator é utilizado como divisor na fórmula, invertendo a lógica para que um valor menor no denominador resulte em maior severidade.

Tabela 5.1: Fatores considerados no cálculo da severidade.

Fator	Condição	Valor
estadoContaScore	Conta ativa	0.5
	Conta bloqueada/expirada	0.3
tipoContaScore	Utilizador sem privilégios de administração	0.1
	Utilizador com privilégios de administração	0.5
acessosScore	Acessos após saída	1
	Sem acessos após saída	1.5

A severidade é calculada utilizando a seguinte equação:

$$Severidade = \frac{10 \times (estadoContaScore + tipoContaScore)}{acessosScore} \quad (5.1)$$

Desta forma, contas ativas de utilizadores privilegiados com acessos após a saída apresentam severidade máxima (10), enquanto contas bloqueadas ou expiradas, de utilizadores comuns, sem acessos apresentam severidade mínima (2.7).

Exemplos de cálculo

- **Conta ativa, com privilégios de administração e com acessos após saída:**

$$Severidade = \frac{10 \times (0.5 + 0.5)}{1} = 10 \quad (5.2)$$

- **Conta bloqueada, sem privilégios de administração e sem acessos após saída:**

$$Severidade = \frac{10 \times (0.3 + 0.1)}{1.5} = 2.7 \quad (5.3)$$

Estes resultados mostram que valores mais elevados indicam contas com maior severidade. Por exemplo, uma severidade de 10 representa um impacto significativamente maior do que uma severidade de 2.7.

Reporting dos Casos de *HouseKeeping* BD

O subprocesso final tem como objetivo a geração e envio automático do relatório dos casos identificados para *HouseKeeping* BD. Este *reporting* é realizado periodicamente, numa data fixa definida no calendário de execução do processo, garantindo que a informação é enviada via e-mail de forma consistente e atempada aos responsáveis hierárquicos do titular da conta. Na ausência de um responsável hierárquico atribuído, o relatório é direcionado ao gestor de serviço da base de dados onde a conta está criada.

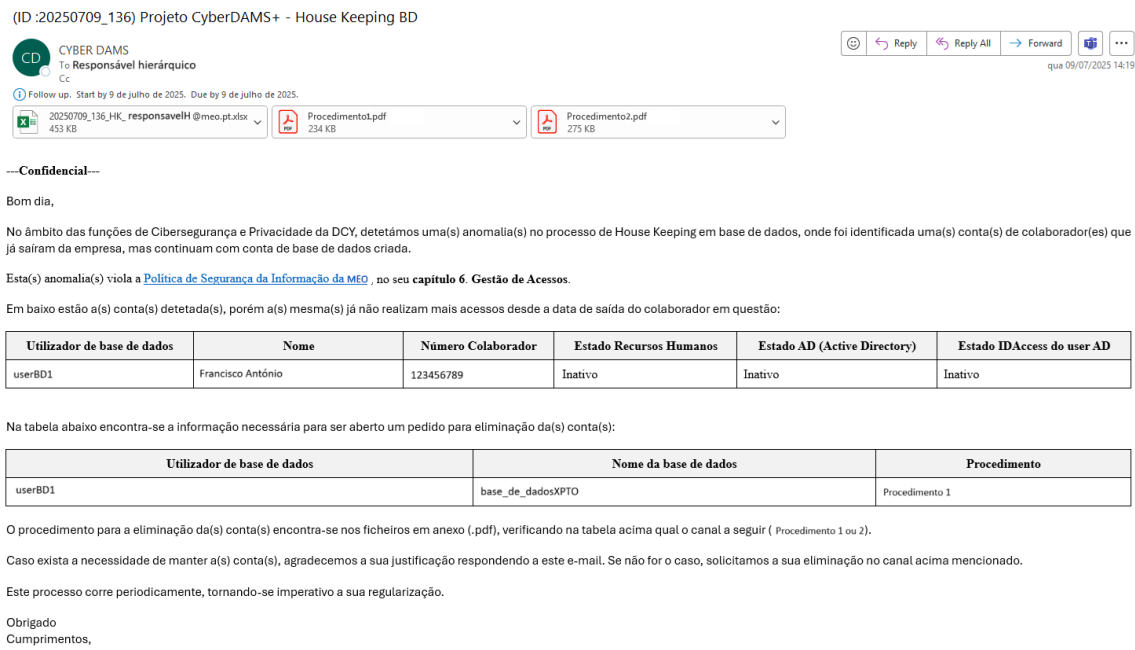
No corpo do e-mail (um exemplo na Figura 5.5) é apresentada uma síntese dos principais casos identificados, destacando as informações relativas à conta partilhada e às contas que estão a usar a mesma, bem como o procedimento que o destinatário deste e-mail deve proceder.

Os ficheiros PDF relativos aos procedimentos apresentam, de forma resumida, os passos a seguir, acompanhados de capturas de ecrã que ilustram como o utilizador deve proceder para corrigir a não conformidade detetada.

Em anexo, segue um ficheiro Excel contendo:

- **Lista detalhada de casos:** cada linha corresponde a uma conta sinalizada, incluindo campos como o identificador do utilizador, sistema, data de saída, estado da conta, existência de acessos após saída, severidade calculada e outras informações relevantes.
- **Evidências associadas:** datas e horários de acessos, origem dos acessos (máquina/IP), nome da base de dados e ações registadas.

Este mecanismo garante que todos os intervenientes têm acesso a informação clara, estruturada e acionável, permitindo agir rapidamente sobre contas que representem riscos de segurança.



É importante destacar a visão detalhada do processo, que sintetiza todos os subprocessos descritos anteriormente e a forma como estes interagem entre si. Essa representação encontra-se ilustrada na Figura 5.6.

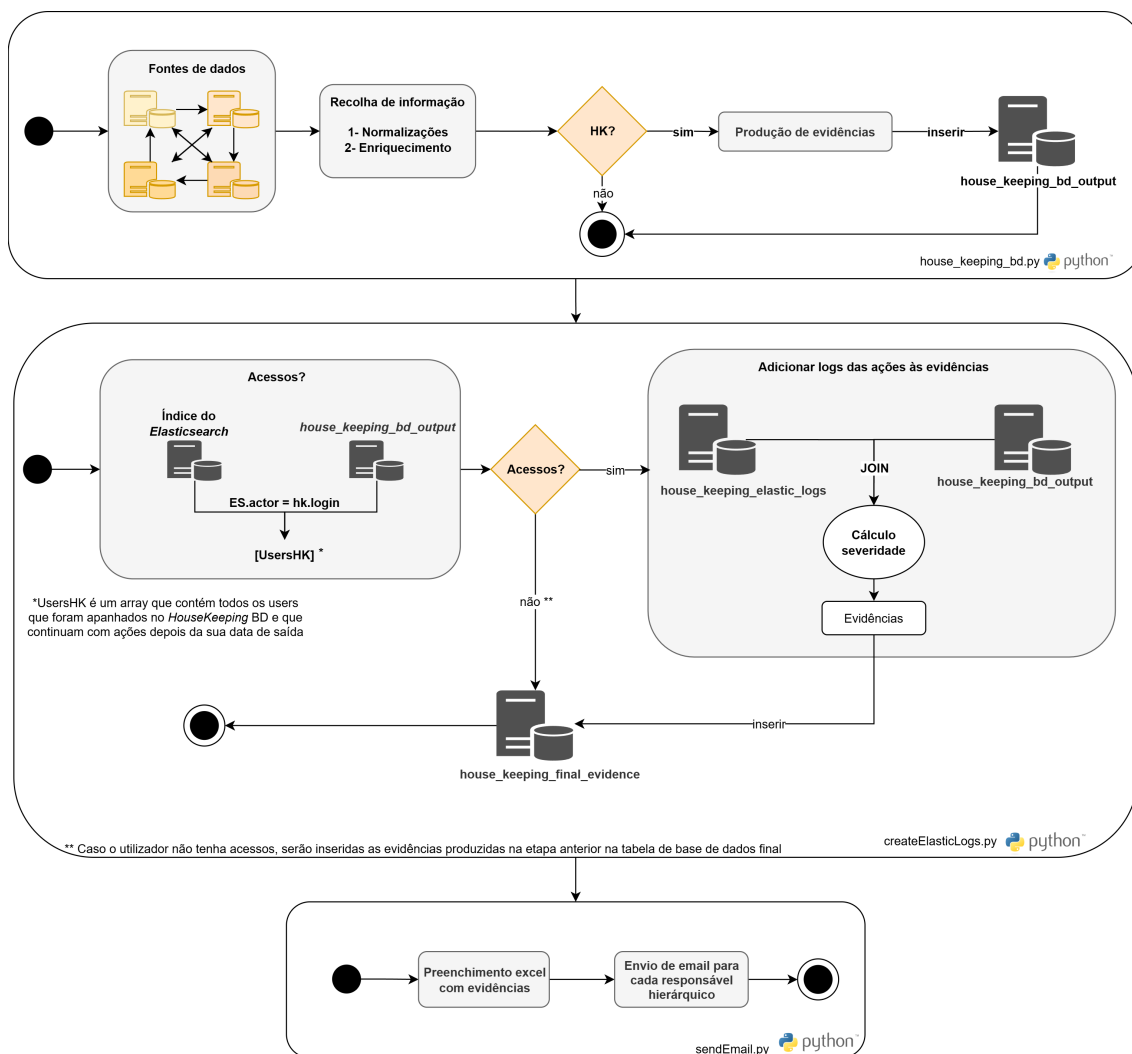


Figura 5.6: Visão detalhada do processo de *HouseKeeping* BD.

Testes realizados

Para validar a veracidade e a robustez do processo implementado, foram realizados testes controlados que incluíram a manipulação deliberada das fontes de dados. O objetivo foi simular cenários que o processo deveria identificar e reportar como não conformidades.

No ambiente de desenvolvimento, a base de dados **ETLS** foi intencionalmente alterada, com a inserção de utilizadores que já tinham saído da empresa, abrangendo todas as combinações possíveis. Por exemplo, criou-se um caso em que um utilizador interno, que já saiu da organização e marcado como inativo na fonte de dados de RH, mantinha ainda uma conta de base de dados ativa, reproduzindo assim uma situação real de incumprimento.

Antes de apresentar os resultados, importa referir que as fontes de dados não têm o mesmo peso na decisão final. Existe uma hierarquia de prioridade definida de acordo com a fiabilidade e precisão de cada fonte: EXCEP > RH > AD > GI.

Nas tabelas 5.2 e 5.3, a sigla NA indica *Não Aplicável*, significando que o estado da fonte de dados (Ativa ou Inativa) é irrelevante para o contexto em questão. O símbolo “-” representa a ausência de informação na fonte de dados relativamente ao seu estado.

Tabela 5.2: Combinações a reportar.

Estado GI	Estado AD	Estado RH	Estado EXCEP	Combinação
NA	NA	Inativo	Inativo	RH Inativo
NA	Inativo	Inativo	Inativo	AD Inativo
Inativo	-	-	-	GH Inativo

Tabela 5.3: Combinações a não reportar.

Estado GI	Estado AD	Estado RH	Estado EXCEP	Combinação
NA	NA	NA	Ativa	Exceção Ativa
NA	NA	Ativo	Inativo	RH Ativo
NA	Ativo	Inativo	Inativo	AD Ativo
Ativo	-	-	-	GH Ativo

Observando as tabelas, percebe-se a lógica da prioridade: por exemplo, na Tabela 5.3, sempre que EXCEP está ativo, as restantes colunas podem estar vazias ou marcadas como NA, pois a exceção sozinha basta para não acionar o processo de HK. De forma análoga, quando RH indica “Ativo”, as situações de AD ou GI tornam-se secundárias.

Após a execução de todos os testes simulados, o processo identificou corretamente todos os casos através do mecanismo de *HouseKeeping* BD, confirmando que o sistema opera como esperado e deteta incoerências entre o estado real das identidades e o seu envolvimento nos sistemas.

Resultados

A implementação do processo de *HouseKeeping* BD permitiu identificar, em ambiente real, contas de colaboradores já desligados da organização que permaneciam ativas em bases de dados. Embora os testes controlados anteriores tenham demonstrado 100% de eficácia na deteção de situações de risco, o foco desta fase recaiu sobre a identificação e tratamento de casos efetivos, permitindo atuar diretamente na mitigação de potenciais falhas de segurança.

Importa salientar que o número total de contas a serem monitorizadas corresponde apenas a uma amostra representativa do universo global de contas de base de dados existentes na organização.

Assim, os resultados apresentados refletem as ocorrências detetadas neste subconjunto, funcionando como indicadores relevantes para aferir a eficácia e aplicabilidade do processo, e não como valores absolutos para toda a infraestrutura.

Os principais resultados obtidos incluem:

- Identificação de contas de BD existentes associadas a colaboradores que já saíram, representando 7% face ao número total de contas a serem monitorizadas no âmbito deste processo;
- Identificação de contas identificadas para *HouseKeeping BD* que realizam ações, correspondendo a 0,1% do número total de contas a serem monitorizadas no âmbito deste processo;
- Identificação de um falso positivo;
- Categorização das contas com base em critérios objetivos de risco, permitindo a atribuição de índices de severidade que facilitam a priorização das ações de remediação;
- Geração automática de relatórios consolidados com evidência clara das anomalias detetadas, promovendo maior agilidade na tomada de decisões;
- Validação do processo através de testes controlados, que confirmaram a robustez e fiabilidade da lógica implementada, mesmo em cenários simulados com dados manipulados.

O falso positivo identificado no processo corresponde ao caso em que, devido a limitações técnicas de alguns sistemas relacionadas com os nomes de utilizador (como comprimento máximo ou utilização de caracteres especiais), determinados colaboradores não podem utilizar, na base de dados, o mesmo identificador que têm atualmente no *Active Directory*. Nestes casos, recorre-se a *logins* alternativos, o que leva o processo de *HouseKeeping BD* a detetar incorretamente a conta como estando associada a um colaborador inativo. Para mitigar este problema, foi criada uma tabela de exceções onde estas situações são registadas e justificadas, garantindo que são ignoradas nas execuções subsequentes do processo.

Na Tabela 5.4, apresenta-se a distribuição percentual das principais combinações de casos reais detetados pelo processo, evidenciando a frequência com que determinadas fontes de dados indicam contas em situação de risco:

Tabela 5.4: Origem dos casos detetados.

Fonte que indicou inatividade	Percentagem de casos
Plataforma GI	73%
Recursos Humanos (RH)	14%
Active Directory (AD)	13%

Dashboards

Para facilitar a monitorização contínua e a análise visual dos dados recolhidos, foram desenvolvidos *dashboards* dinâmicos e interativos que resumem os principais indicadores de desempenho e risco associados às contas analisadas.

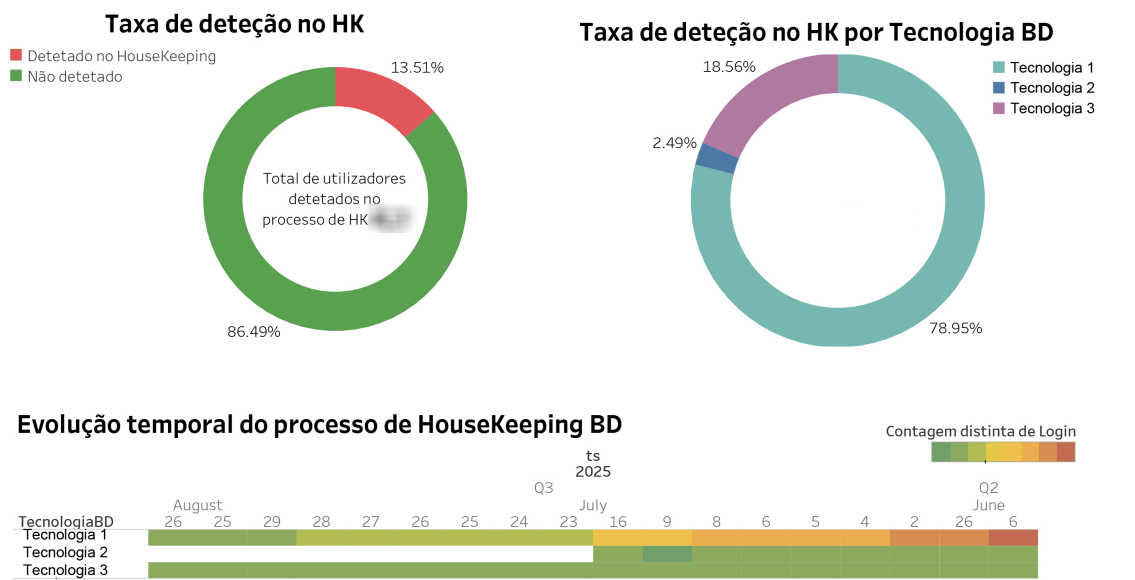


Figura 5.7: *Dashboards* do processo de *HouseKeeping* BD que contém métricas como a taxa de deteção, taxa de deteção por tecnologia de BD e uma *timeline* que mostra o número de casos detetados ao longo do tempo.

Taxa de utilizadores detetados no HouseKeeping BD que estão ativos

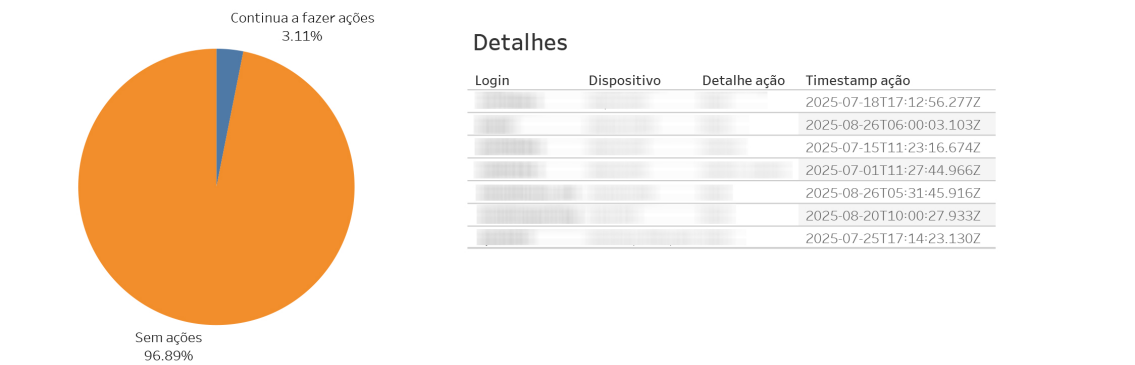


Figura 5.8: Casos detetados no processo de *HouseKeeping* BD que continuam a realizar ações após a data de saída do colaborador da organização.

Nota: Todos os dados apresentados nas figuras foram anonimizados para garantir a privacidade.

A Figura 5.7 apresenta um *dashboard* que contém uma *timeline* que regista o histórico de todas as execuções do projeto, cobrindo o período de junho a agosto. A legenda de cores indica a quantidade de utilizadores detetados: laranja representa o valor máximo observado, enquanto o verde corresponde aos valores mais baixos. A análise do *dashboard* permite observar que, ao longo do período, a quantidade de contas nesta situação diminuiu. Em particular, durante o mês de agosto, a cor na *timeline* passa de tons vermelhos ou laranja para verde, indicando que houve uma redução significativa no número de utilizadores detetados nesta situação. Inclui ainda dois gráficos circulares: o primeiro representa a taxa de deteção de utilizadores no processo de *HouseKeeping BD*, e o segundo mostra a distribuição dessa deteção por tecnologia de base de dados. Ambos os gráficos consideram apenas os dados da última execução como filtro.

Estes *dashboards* permitem o acompanhamento em tempo real do estado das contas e a sua evolução ao longo do tempo.

Conclusão

Como conclusão, o processo de *HouseKeeping BD* revelou-se essencial para reforçar a segurança dos sistemas corporativos, ao garantir que não subsistem contas sem responsável ou indevidamente ativas após a saída de colaboradores. A sua integração numa rotina periódica de análise promove não só a conformidade com boas práticas de segurança, mas também uma maior governança sobre os acessos aos ativos críticos da organização.

A continuidade deste processo, com iterações regulares e eventuais melhorias na precisão das fontes de dados e mecanismos de cruzamento contribuirá significativamente para a mitigação de riscos operacionais e de segurança associados à gestão de identidades em ambientes de base de dados.

5.2.2 Partilha de Contas

O processo de deteção de partilha de contas encontra-se dividido em dois subprocessos principais, cada um com um objetivo e execução distintos.

Na Tabela 5.5 são apresentados alguns campos que vão ser analisados na deteção de partilha de contas.

Tabela 5.5: Campos dos logs de base de dados.

Campos	Definição
<i>dbuser</i>	Utilizador de base de dados
<i>dbuserWorkerID</i>	Número mecanográfico do <i>dbuser</i>
<i>osuser</i>	Utilizador de sistema utilizado no acesso ao <i>client name</i>
<i>osuserWorkerID</i>	Número mecanográfico do <i>osuser</i>
<i>clientName</i>	Designação da máquina utilizada para realizar o acesso à base de dados
<i>clientIP</i>	Endereço IP do <i>clientName</i>
<i>database</i>	Nome da base de dados acedida
<i>instance</i>	Instância da base de dados
<i>serverName</i>	Designação do servidor da base de dados
<i>serverIP</i>	Endereço IP do servidor
<i>serverPort</i>	Porta do servidor
<i>program</i>	Programa utilizado para aceder à base de dados
<i>ts</i>	Timestamp em que foi realizado o acesso
<i>FullSQL</i>	Comando SQL usado
<i>External ID</i>	ID único referente a uma ação
<i>actor_range_type</i>	Localização do acesso

Definição dos cenários de Partilha de Contas

O primeiro subprocesso visa identificar situações de partilha indevida de contas, com base na análise dos cenários implementados.

Inicialmente, a fonte de dados GI é consultada de forma a extrair as contas nominais e aplicacionais. Em paralelo, são identificados os utilizadores que executaram ações nos últimos 30 dias, com base nos dados recolhidos num índice do *Elasticsearch* que contém os logs de acesso a BDs. Após a recolha, os dados são normalizados, o que inclui a remoção de *dbusers* nulos e a exclusão de registos onde o *dbuser* coincide com o *osuser*.

Antes de proceder à análise dos cenários de partilha de contas, importa apresentar o cenário ideal de utilização de credenciais, que constitui a referência para a deteção de não conformidades. Neste modelo, cada utilizador utiliza exclusivamente a sua conta nominal, e as contas aplicacionais são reservadas apenas para sistemas e integrações automáticas.

As figuras [5.9](#) e [5.10](#) ilustram este cenário de conformidade, no qual não há partilha de contas.

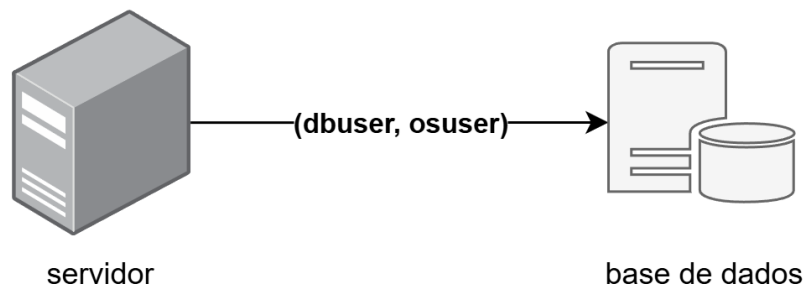


Figura 5.9: Cenário correto de um utilizador aplicacional aceder a uma conta de base de dados aplicacional.

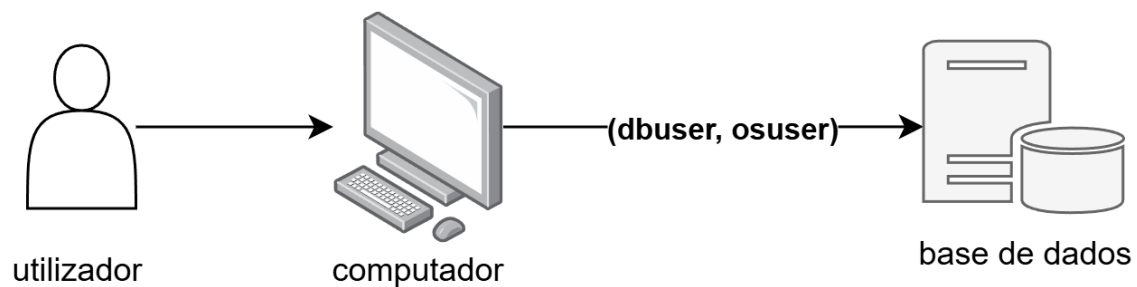


Figura 5.10: Cenário correto de um utilizador nominal aceder a uma conta de base de dados nominal.

No cenário apresentado na Figura 5.9, o acesso é efetuado por um processo que corre num servidor e comunica com a base de dados (por exemplo, um processo automático). Esse acesso pode também ocorrer através de uma aplicação, como uma página web que executa consultas à base de dados. Nessa situação, não é o utilizador com a sua conta pessoal que executa as consultas, mas sim a conta aplicacional associada ao serviço, da qual o utilizador é responsável. Neste primeiro cenário, tanto o `dbuser` como o `osuser` correspondem a utilizadores aplicacionais, sendo o servidor o `osuser` aplicacional e a conta de base de dados o `dbuser` aplicacional.

Já a Figura 5.10 representa um acesso direto a uma base de dados realizado por um utilizador, a partir do seu próprio computador, utilizando a sua conta nominal. Aqui, o `dbuser` e o `osuser` são ambos nominais, sendo o computador o `osuser` nominal (conta nominal do utilizador) e a conta de base de dados o `dbuser` nominal.

Com os dados devidamente preparados, procede-se à análise dos cinco cenários de partilha de contas, apresentados na tabela VII.

Tabela 5.6: Cenários implementados para detecção de partilha de contas.

cenário	<i>dbuser</i>	<i>osuser</i>	computador	IP
nom-nom	user nominal A	user nominal B	-	-
app-nom	user aplicacional A	user nominal B	-	-
nom-pc	user nominal A	-	computador B	-
app-pc	user aplicacional A	-	computador B	-
app-vpn	user aplicacional A	-	-	vpn (nominal)

O **cenário nom-nom** ocorre quando existe utilização indevida de credenciais nominais, podendo manifestar-se de duas formas: quando vários *osusers* nominais acedem ao mesmo *dbuser* nominal, ou quando um único *osuser* nominal acede a um *dbuser* nominal que não lhe pertence.

O **cenário app-nom** é igual ao cenário anterior, porém o acesso ao *dbuser* nominal é feito por *osuser* aplicacional.

O **cenário nom-pc** diz respeito à associação entre *dbusers* nominais e os computadores a partir dos quais efetuam os acessos. Quando um mesmo utilizador acede ao sistema a partir de diferentes dispositivos, pode-se inferir a possibilidade de utilização de máquinas que não lhe são atribuídas.

O **cenário app-pc** é semelhante ao anterior, porém apenas é monitorizado a nomenclatura do computador cujo acesso foi feito, caso um acesso via utilizador aplicacional contenha uma nomenclatura típica de estações de trabalho de utilizadores nominais, evidencia um acesso idêntico ao **cenário app-nom**, o que é uma não conformidade.

O **cenário app-vpn** é relacionado com o uso indevido de *dbusers* aplicacionais detetado via IPs de VPN. Este ocorre quando uma conta de base de dados do tipo aplicacional é utilizada através de um IP de VPN, evidenciando um acesso idêntico ao **cenário app-nom**.

A identificação de qualquer um destes cinco cenários é suficiente para concluir que existe partilha de contas. Caso contrário, isto é, na ausência de evidência destes padrões de comportamento, considera-se que não há indícios de partilha e que os princípios de rastreabilidade e segurança estão a ser respeitados neste âmbito.

Deteção de Partilha de Contas

No cenário nom-nom, tanto o *dbuser* como o *osuser* são contas nominais. Aqui, são comparados os pares extraídos do *Elasticsearch* com a lista de utilizadores nominais, com o objetivo de verificar se ambos pertencem à mesma pessoa. Esta verificação passa por verificar na fonte de dados GI qual o id de identidade de cada conta, e verificar se ambos os ids são iguais, se forem iguais significa que é a mesma identidade/pessoa, caso contrário tratam-se de identidades diferentes, visto que uma pessoa apenas tem uma identidade. Quando estas contas nominais estão associadas a identidades diferentes, configura-se uma situação de anomalia, reveladora de partilha indevida de contas entre indivíduos distintos.

O cenário app-nom analisa situações em que o *dbuser* é aplicacional e o *osuser* é nominal.

Para tal, procede-se à correspondência de cada elemento com as listas respetivas de contas aplicacionais e nominais. Uma conta de base de dados aplicacional não deve ser acedida por uma conta nominal, e, idealmente, deve ser acedida exclusivamente por serviços aplicacionais (tipicamente associadas a servidores). Assim, mesmo que um acesso direto a uma base de dados seja realizado por um utilizador aplicacional, mas executado por uma pessoa, a situação é considerada uma não conformidade. Por outro lado, o cenário inverso, *dbuser* nominal acedido por um *osuser* aplicacional, não constitui partilha de conta, dado que pode decorrer, por exemplo, da execução de uma tarefa por um programador a partir de um servidor.

O cenários nom-pc, app-pc e app-vpn contemplam situações em que o campo *osuser* está ausente. Estas podem ocorrer devido a falhas na recolha ou preenchimento de dados no *Elasticsearch*. Nesses casos, a deteção de partilha de contas depende da análise de atributos complementares, como o computador usado no acesso ou o IP de origem.

No cenário nom-pc, quando o *dbuser* é nominal e o *osuser* está em falta, a análise centra-se na identificação do responsável pelo computador utilizado na ação. Se este for o mesmo que o titular da conta de base de dados, a situação é considerada regular. Caso contrário, é verificado se ambas as identidades correspondem à mesma pessoa. Quando não há correspondência, considera-se uma anomalia, pois implica que uma conta nominal está a ser usada por um terceiro.

No cenário app-pc, quando o *dbuser* é do tipo aplicacional e o *osuser* está em falta, podem ser identificadas anomalias se o nome do computador corresponder a padrões típicos de dispositivos atribuídos a utilizadores nominais. Caso essa análise não permita detetar não conformidades, é efetuada uma segunda verificação, correspondente ao cenário app-vpn. Nesta verificação, analisa-se o endereço IP para confirmar se se trata de uma ligação VPN. Se for o caso, tal indica a possibilidade de acesso remoto à conta *dbuser* aplicacional a partir de um dispositivo pessoal.

Após a análise de todos os casos, os resultados são agregados, removendo-se os duplicados com base no identificador externo (*externalID*). Para além da identificação das anomalias, são recolhidas informações complementares para efeitos de *reporting*, incluindo o departamento, direção, nome e número mecanográfico dos utilizadores envolvidos (*osuser* e *dbuser*), bem como detalhes adicionais das ações registadas no *Elasticsearch*. É igualmente identificada a cadeia hierárquica dos responsáveis por cada utilizador.

Cálculo da Severidade

Tal como no caso de uso “*HouseKeeping BD*”, apesar de todas as conformidades terem de ser corrigidas, é calculada a severidade para priorizar a remediação das contas de base de dados partilhadas. Com base nos dados recolhidos, calcula-se a severidade de cada ocorrência, considerando fatores como o número de utilizadores diferentes que acederam a uma mesma conta de base de dados (*dbuser*) e o volume de ações críticas ou não críticas realizadas. Neste âmbito, uma ocorrência é um caso de partilha de conta, onde ocorre pelo menos um dos cenários apresentados anteriormente. Por exemplo, se uma conta de base de dados for acedida por três utilizadores diferentes,

e cada um deles fizer 20 ações críticas e 5 ações não críticas, então a severidade será calculada tendo em conta esses três parâmetros.

A severidade é determinada a partir de três fatores que avaliam o potencial impacto da partilha de contas:

- **numeroUtilizadoresScore:** representa quantos utilizadores distintos acederam ao mesmo *dbuser*. Quanto mais utilizadores diferentes partilharem a conta, maior o potencial de exposição e, portanto, maior o valor atribuído;
- **numeroLoginsScore:** reflete o volume total de *logins* realizados. Um maior número de *logins* amplia a exposição da conta, pois indica uso mais frequente da conta partilhada;
- **numeroAcoesCriticasScore:** mede a quantidade de ações críticas efetuadas, como operações de administração ou alterações sensíveis na base de dados. Este fator tem o maior peso, dado que ações críticas representam o maior potencial de impacto.

A atribuição dos *scores* segue uma escala linear, com incrementos de 0,25. Assim, valores mais altos indicam maior criticidade:

- valores até cerca de 0,25 refletem baixa relevância;
- em torno de 0,50 representam relevância média;
- 1,00 corresponde ao nível máximo, considerado crítico.

As tabelas 5.7, 5.8 e 5.9 apresentam os escalões de cada fator:

Tabela 5.7: Atribuição do *numeroUtilizadoresScore*.

Nº de utilizadores	numeroUtilizadoresScore
1	0.25
2	0.50
3	0.75
≥ 4	1.00

Tabela 5.8: Atribuição do *numeroLoginsScore*.

Nº de logins	numeroLoginsScore
1 – 10	0.25
11 – 50	0.50
51 – 200	0.75
≥ 201	1.00

Tabela 5.9: Atribuição do *numeroAcoesCriticasScore*.

Nº de ações críticas	<i>numeroAcoesCriticasScore</i>
1 – 5	0.25
6 – 25	0.50
26 – 50	0.75
≥ 51	1.00

A severidade é então calculada pela seguinte expressão:

$$Severidade = (0.5 \times numeroAcoesCriticasScore + 0.25 \times numeroLoginsScore + 0.25 \times numeroUtilizadoresScore) \times 10 \quad (5.4)$$

Desta forma o valor mínimo de severidade é 1.25 (quando não foram efetuadas ações críticas, existe apenas um *login* e apenas um utilizador) e o valor máximo de severidade é 10 (quando todos os scores são o máximo estipulado).

Exemplos de cálculo

- **Conta de base de dados acedida por 5 utilizadores, registou 300 *logins* e efetuou 60 ações críticas:**

$$Severidade = (0.5 \times 1 + 0.25 \times 1 + 0.25 \times 1) \times 10 = 10 \quad (5.5)$$

- **Conta de base de dados acedida por 1 utilizadores, registou 23 *logins* e efetuou 0 ações críticas:**

$$Severidade = (0.5 \times 0.25 + 0.25 \times 0.5 + 0.25 \times 0) \times 10 = 2.5 \quad (5.6)$$

Estes resultados ilustram que valores altos de severidade indicam contas mais críticas e prioritárias para remediação. Por exemplo, uma severidade de 10 corresponde a uma ocorrência de elevado impacto, enquanto uma severidade de 2.5 indica uma ocorrência de impacto menor.

Reporting dos users

O objetivo do último subprocesso é efetuar o reporting automático dos casos apanhados de partilha de contas. O processo começa por preencher um *template* de excel com as evidências da tabela anterior, e afixá-lo ao e-mail de *reporting* gerado.

A lógica do *reporting* consiste em enviar um relatório individual para cada responsável hierárquico, contendo todos os *dbusers* sob a sua responsabilidade, juntamente com os respetivos *osusers* que utilizaram essas contas, ou responsáveis das máquinas que acederam à mesma,

dependendo de qual cenário foi detetado. Na eventualidade do *dbuser* em questão não ter nenhum responsável hierárquico associado, o caso é reportado para o gestor de serviço da base de dados onde aconteceu a partilha de contas.

O e-mail de *reporting* inclui duas tabelas por cada caso de partilha de conta. A tabela 5.10 apresenta a conta de base de dados em partilha.

Tabela 5.10: Informação da conta de base de dados detetada em partilha.

Conta de rede	Nome	Direção	Departamento	Nº Mecanográfico
userBD1	Jorge António	Direcao5	Departamento5	12345

A segunda apresenta todas as contas de rede que acederam a essa conta, representada na Tabela XII, ou, nos casos em que não exista informação sobre o utilizador do sistema operativo (*osuser*), os dados dos computadores que realizaram o acesso, representada na Tabela XIII.

Tabela 5.11: Contas de rede que acederam à conta de base de dados partilhada.

Conta de rede	Nome	Direção	Departamento	Nº Mecanográfico	Base de dados
user01	Francisco António	Direcao1	Departamento1	98765	BD1
user02	Tomás Fernandes	Direcao2	Departamento2	12345	BD2

Tabela 5.12: Computadores que acederam à conta de base de dados partilhada (quando não existe informação de *osuser*).

Computador usado	IP do computador	Programa	Base de dados
PC01	10.0.0.0	SQLCLIENT	BD5
PC02	10.0.0.1	SQLCLIENT	BD4

A Figura 5.2 apresenta uma visão detalhada do processo de deteção de partilha de contas. Nela estão representados os subprocessos descritos anteriormente. Esta representação fornece uma perspetiva integrada e facilita a compreensão da forma como os vários componentes do processo se interligam para garantir a deteção eficaz e sistemática de situações de não conformidade.

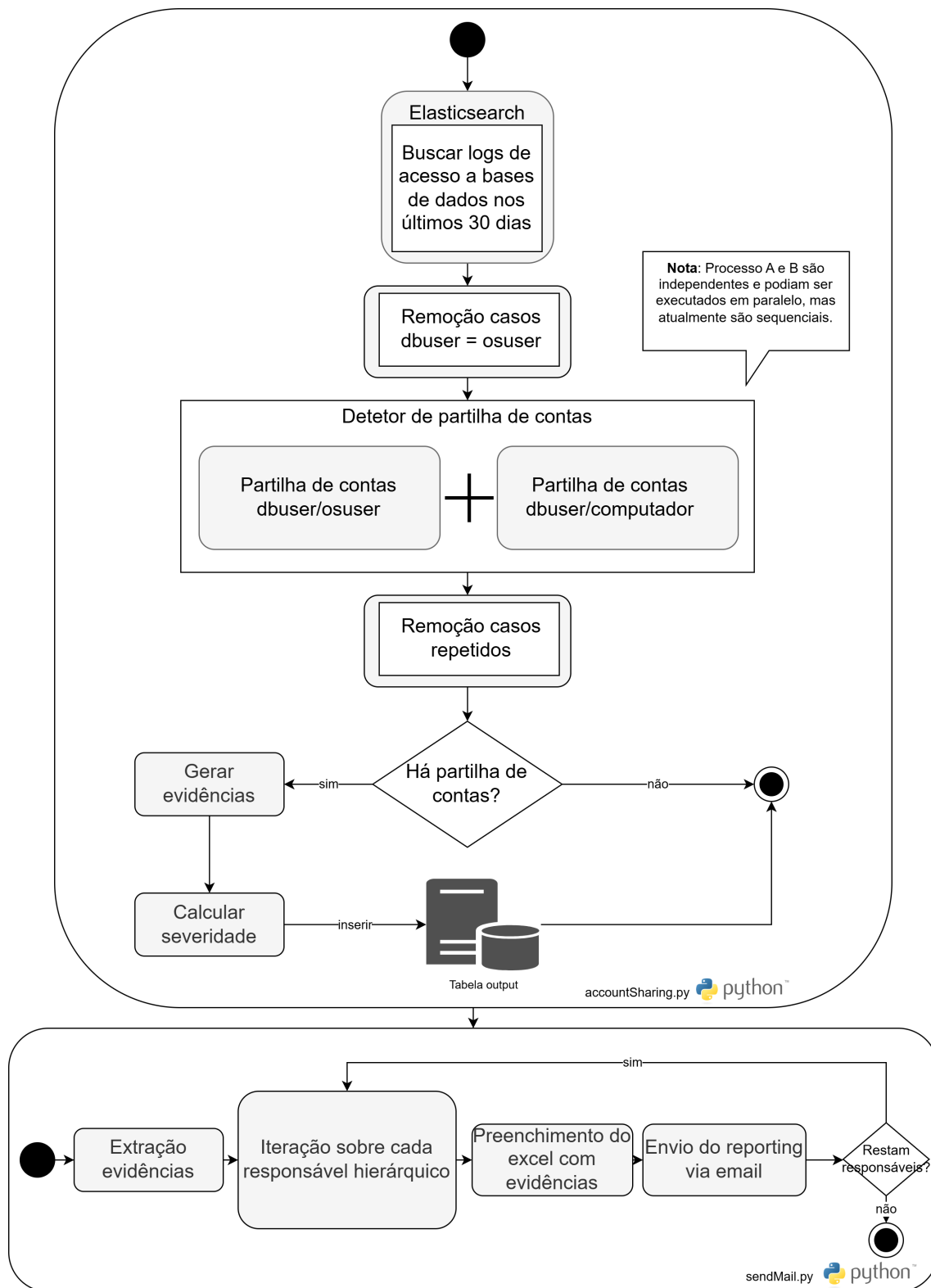


Figura 5.11: Arquitetura Micro da Partilha de Contas.

Resolução do Problema

O processo de detecção de partilhas de contas será executado mensalmente, no dia 1 de cada mês. O relatório correspondente será enviado no dia 1 do mês seguinte, às 08h00. Este mecanismo garante uma monitorização sistemática e contínua, assegurando que todas as ocorrências de partilha indevida de contas são devidamente reportadas e acompanhadas até à sua completa resolução.

Importa salientar que a gravidade de cada caso de partilha de contas pode variar significativamente. Por exemplo, situações em que duas pessoas da mesma equipa recorrem temporariamente à mesma conta por motivos técnicos tendem a apresentar um risco reduzido. Em contraste, cenários de partilha envolvendo acessos *nom-nom* ou, de forma ainda mais crítica, casos em que um acesso *app-nom* é utilizado para a extração de dados sensíveis representam riscos muito mais severos e exigem uma resposta imediata.

Devido a esta diferenciação de gravidade, não é possível definir previamente uma resolução única e padronizada para todos os cenários. A decisão sobre a resposta adequada só pode ser tomada após a análise detalhada de cada ocorrência e da respetiva justificação recebida no âmbito do processo de *reporting*.

Após o envio do relatório, cabe ao responsável analisar as situações reportadas. Essa análise deve determinar se a partilha de contas detetada é justificada (e devidamente documentada) ou se configura uma não conformidade. A resposta pode consistir em:

- **Justificação formal da situação**, no caso de ser uma exceção válida e autorizada;
- **Ação corretiva imediata**, nos casos em que a partilha não é aceitável.

Testes realizados

Para validar os diferentes cenários definidos, foi necessário realizar apenas dois acessos correspondentes a partilha de contas. No primeiro, utilizei o meu próprio computador para aceder à minha conta de base de dados nominal, e, em seguida, a mesma conta foi acedida a partir de outro computador com um utilizador de rede distinto. Este procedimento corresponde ao cenário *nom-nom*.

De forma semelhante, foi realizado um segundo teste, mas desta vez utilizando uma conta de base de dados aplicacional, o que permitiu validar o cenário *app-nom*.

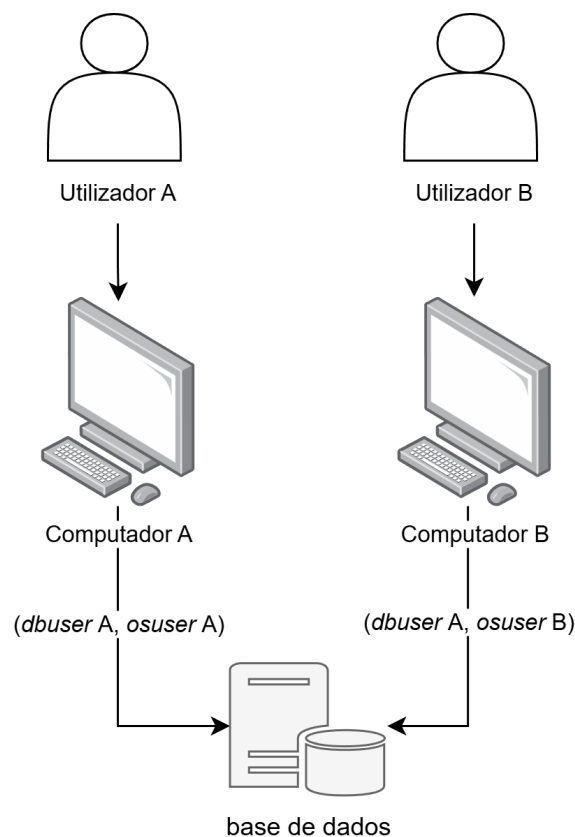


Figura 5.12: Acesso correspondente a partilha de conta.

Para os restantes três cenários (*nom-pc*, *app-pc* e *app-vpn*), a estratégia adotada consistiu em manipular os logs de acessos às bases de dados, efetuados na Figura 5.12 durante o processamento em *Python*, de forma a que o campo *osuser* surgisse em branco. Esta alteração simulou situações em que não existe informação sobre o utilizador do sistema operativo, permitindo assim testar integralmente todos os casos previstos.

Adicionalmente, foram realizados testes à componente de *reporting*, com a seleção manual de casos reais de partilha de contas. Esses casos foram reportados a responsáveis hierárquicos através do processo automatizado, permitindo avaliar a clareza e eficácia do formato de comunicação. Com base no *feedback* obtido, foram introduzidas melhorias no *layout* e no conteúdo dos relatórios enviados.

Os testes demonstraram que a lógica de deteção funciona conforme o esperado e que o mecanismo de *reporting* é eficaz na comunicação de situações de risco, contribuindo para a melhoria contínua do processo de gestão de acessos.

Resultados

A fase de testes do processo de deteção de partilha de contas permitiu validar a eficácia do modelo proposto na identificação de situações anómalas em ambientes reais. Os testes abrangeram os cenários definidos.

Importa salientar que o número total de contas a serem monitorizadas corresponde apenas a uma amostra representativa do universo global de contas de base de dados existentes na organização. Assim, os resultados apresentados refletem as ocorrências detetadas neste subconjunto, funcionando como indicadores relevantes para aferir a eficácia e aplicabilidade do processo, e não como valores absolutos para toda a infraestrutura.

Os principais resultados obtidos durante os testes incluem:

- Identificação de casos de partilha de contas distribuídos pelos cenários definidos, sendo que aproximadamente 99% dos casos correspondem ao cenário onde o *dbuser* é nominal e o *osuser* aplicativo;
- Atribuição de uma severidade mínima de 2.5 e máxima 6.25 dos casos detetados;
- Consolidação de indicadores operacionais e de segurança em *dashboards* interativos, facilitando uma análise rápida e eficaz por parte das equipas de segurança e compliance;
- Confirmação da consistência e robustez do processo de deteção, que demonstrou desempenho adequado em contexto de dados reais, com capacidade para identificar e isolar exceções legítimas.

Dashboards

Com o objetivo de facilitar a monitorização e a análise contínua das ocorrências de partilha de contas, foram desenvolvidos *dashboards* interativos que integram os principais indicadores de segurança e conformidade associados a este processo. Estes *dashboards* permitem à equipa de segurança, compliance e gestão identificar rapidamente áreas de risco, padrões recorrentes e evoluções temporais.

Entre os principais elementos visuais disponíveis nos *dashboards* destacam-se:

- Número total de ocorrências em cada execução, por tipo de cenário (nominal-nominal, aplicativo-nominal, etc.);
- Distribuição de severidade das ocorrências, mostrando a média da mesma;
- Ranking do top 5 *dbusers* com mais partilhas identificadas, com opção de detalhar por *osuser* associado;
- Filtros dinâmicos por tipo de base de dados, programa utilizado, entre outros.

Foi desenvolvido também um *dashboard* semelhante ao apresentado na Figura 5.7 que exibe a tendência temporal, permitindo acompanhar a evolução do número de casos e da severidade ao longo do tempo. Contudo, como não houve tempo para a regularização de um número significativo de contas, não é possível visualizar de forma clara essa evolução no momento de escrita do projeto.

Os *dashboards* são atualizados automaticamente a partir das evidências inseridas no sistema, assegurando que as informações apresentadas reflitam sempre o estado mais recente da organização em termos de gestão de acessos e rastreabilidade.

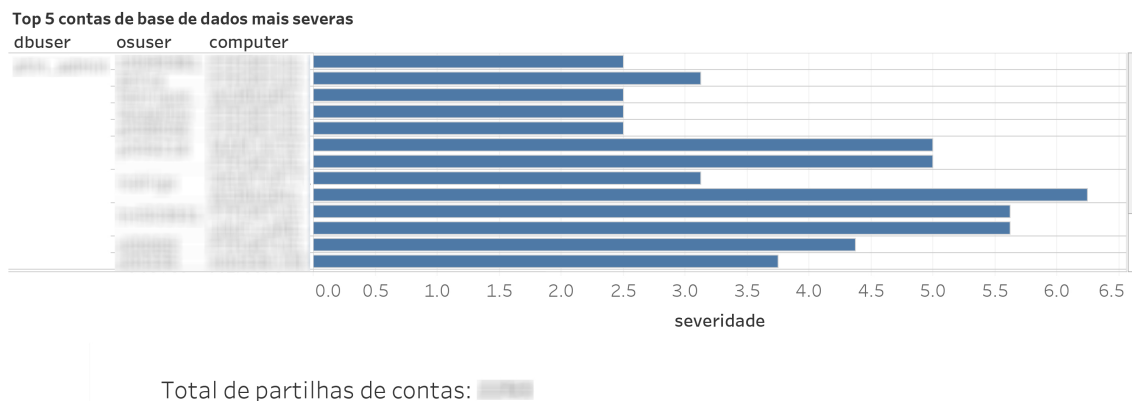


Figura 5.13: *Dashboard* de partilha de contas.

Nota: Todos os dados apresentados nas figuras foram anonimizados para garantir a privacidade.

Conclusão

O processo de deteção e *reporting* de partilha de contas demonstrou ser uma ferramenta eficaz na identificação de situações que comprometem a segurança, integridade e rastreabilidade dos sistemas de informação. A abordagem estruturada, baseada em três cenários de análise distintos, permitiu uma cobertura abrangente dos principais tipos de não conformidade relacionados com a partilha de credenciais.

A severidade atribuída a cada ocorrência proporciona uma priorização clara das situações a resolver, enquanto o mecanismo de *reporting* automático assegura a responsabilização dos gestores diretos e facilita a tomada de decisão rápida e fundamentada.

A introdução de *dashboards* interativos complementa este processo, promovendo a transparência e a capacidade de resposta da organização face a riscos relacionados com a gestão de identidades e acessos.

A fase de testes confirmou a robustez do modelo e a utilidade prática da ferramenta, ao mesmo tempo que revelou oportunidades de melhoria, como o ajuste para falsos positivos. Estes resultados sustentam a continuidade do processo em regime de produção, com execução mensal e monitorização constante, reforçando a postura preventiva da organização no que diz respeito à segurança da informação.

5.2.3 Monitorização DBAs

Os DBAs (*Database Administrators*), ou administradores de bases de dados, são responsáveis pela gestão, configuração e manutenção dos sistemas de base de dados, com privilégios elevados de acesso. O processo de deteção de acessos indevidos a dados sensíveis por parte dos DBAs está estruturado em duas fases complementares: a primeira dedicada à recolha e preparação dos dados; a segunda, à sua monitorização contínua com recurso à plataforma *IBM Guardium*.

Identificação e Recolha de Informações Relativas aos DBAs

A fase inicial consiste na execução de um *script* desenvolvido em *Python*, implementado em *Docker*, responsável por identificar e recolher dados sobre os utilizadores com permissões de administração de bases de dados. Este *script* realiza consultas a uma base de dados de referência, com o objetivo de extrair três elementos principais: a conta de rede do utilizador, a tecnologia associada à base de dados (como *Oracle* ou *SQL Server*, entre outras), e o grupo do AD ao qual o utilizador pertence.

Concluída a recolha, procede-se à normalização das contas de rede. Esta etapa inclui a remoção de domínios e caracteres especiais, bem como a conversão de todos os caracteres para minúsculas, assegurando assim uniformidade nos dados. Esta padronização é essencial para evitar duplicações ou erros decorrentes de variações na formatação dos nomes de utilizador.

Este processo será executado mensalmente, onde os dados tratados são então armazenados numa tabela específica, a qual servirá de base para a fase subsequente de monitorização automatizada.

Monitorização de Acessos dos DBAs a Dados Críticos

Na segunda fase do processo, a ferramenta *IBM Guardium* é utilizada para monitorizar os acessos efetuados pelos DBAs às bases de dados consideradas críticas. Para tal, foram definidos dois grupos funcionais dentro da plataforma: um grupo denominado “DBAs”, que é alimentado automaticamente com os dados extraídos e normalizados na fase anterior, e um grupo designado “Dados Críticos”, previamente existente, que agrega colunas e tabelas com informações sensíveis.

Com base nesses dois grupos, foi configurada uma política de segurança que define a seguinte lógica: sempre que um utilizador pertencente ao grupo “DBAs” executar comandos que não são de administração de BDs a uma tabela que contenha, em qualquer uma das suas colunas, um termo ou expressão incluída no grupo “Dados Críticos”, o sistema deve gerar automaticamente um alarme de segurança.

Este mecanismo permite identificar, em tempo quase real, potenciais acessos indevidos ou não autorizados a dados sensíveis por parte de administradores de bases de dados. Desta forma, assegura-se uma camada adicional de proteção e rastreabilidade no ambiente de bases de dados da organização, em conformidade com os princípios de segurança e governança da informação.

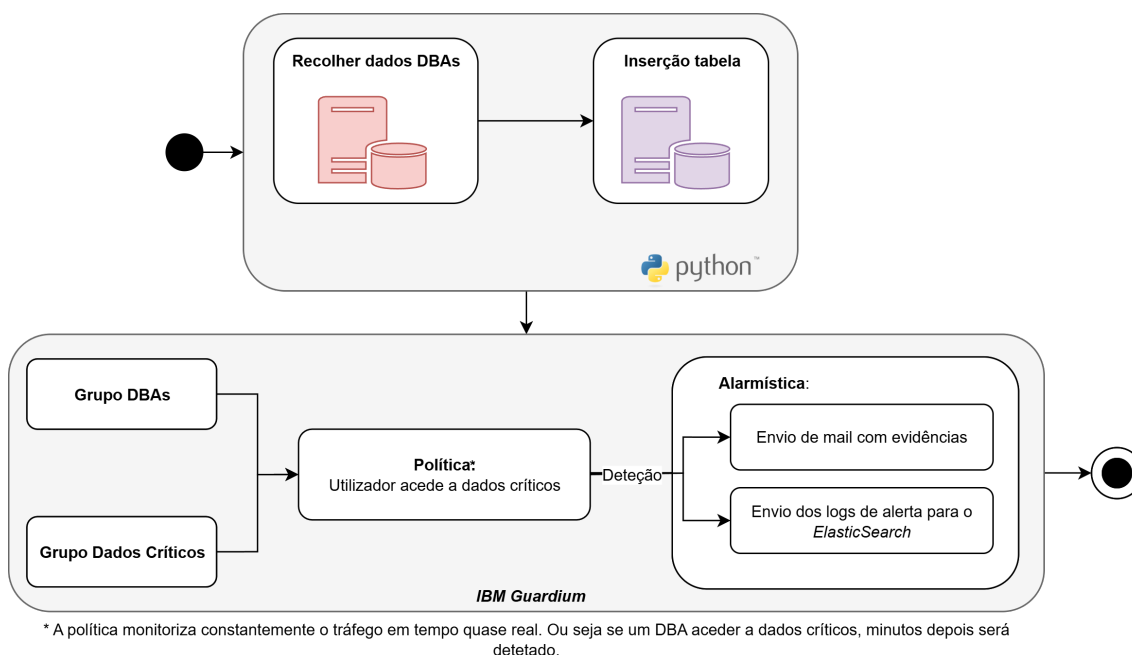


Figura 5.14: Arquitetura micro da Monitorização aos DBAs.

Testes realizados

Foram realizados testes para validar a eficácia do processo de monitorização implementado. Para tal, inseriu-se um utilizador nominal no grupo de DBAs previamente criado na plataforma *IBM Guardium*. Em seguida, foi analisada a tabela que agrega os dados críticos, sendo identificado o campo “Telemovel” como sensível e como objeto de teste.

Posteriormente, foi criada uma base de dados num ambiente controlado de qualidade (QA), contendo uma tabela com uma coluna designada “Telemovel”. Com o utilizador de teste, foi executada uma instrução `SELECT` sobre essa coluna, de forma a verificar se o acesso seria devidamente captado pelo mecanismo de monitorização.

Como resultado, o sistema gerou corretamente um alerta de segurança, confirmando que a política configurada é eficaz na deteção de acessos por parte de DBAs a dados considerados críticos.

Resultados

A monitorização identificou um acesso de um DBA a dados classificados como críticos. Embora o evento correspondesse de facto a um acesso real a informação sensível, este encontrava-se devidamente justificado no âmbito das suas funções de administração de bases de dados. Assim, não se configurando como uma não conformidade, o caso é considerado um falso positivo, uma vez que se tratou de um acesso legítimo, ainda que corretamente assinalado pelo processo de deteção.

Dashboards

Para simplificar a análise e o acompanhamento contínuo dos acessos monitorizados, foram criados *dashboards* no *Elasticsearch*, uma vez que os logs dos acessos detetados pelo *IBM Guardium* são enviados para esta plataforma. Estes dashboards apresentam, de forma visual e interativa, os seguintes elementos principais:

- Lista de acessos realizados por contas classificadas como DBAs a dados críticos;
- Filtros por data, utilizador, base de dados, programa, etc;
- Tendência temporal de alertas de segurança gerados.

Na Figura 5.15 é apresentado um conjunto de *dashboards* que ilustram os resultados obtidos durante os testes realizados a este caso de uso.

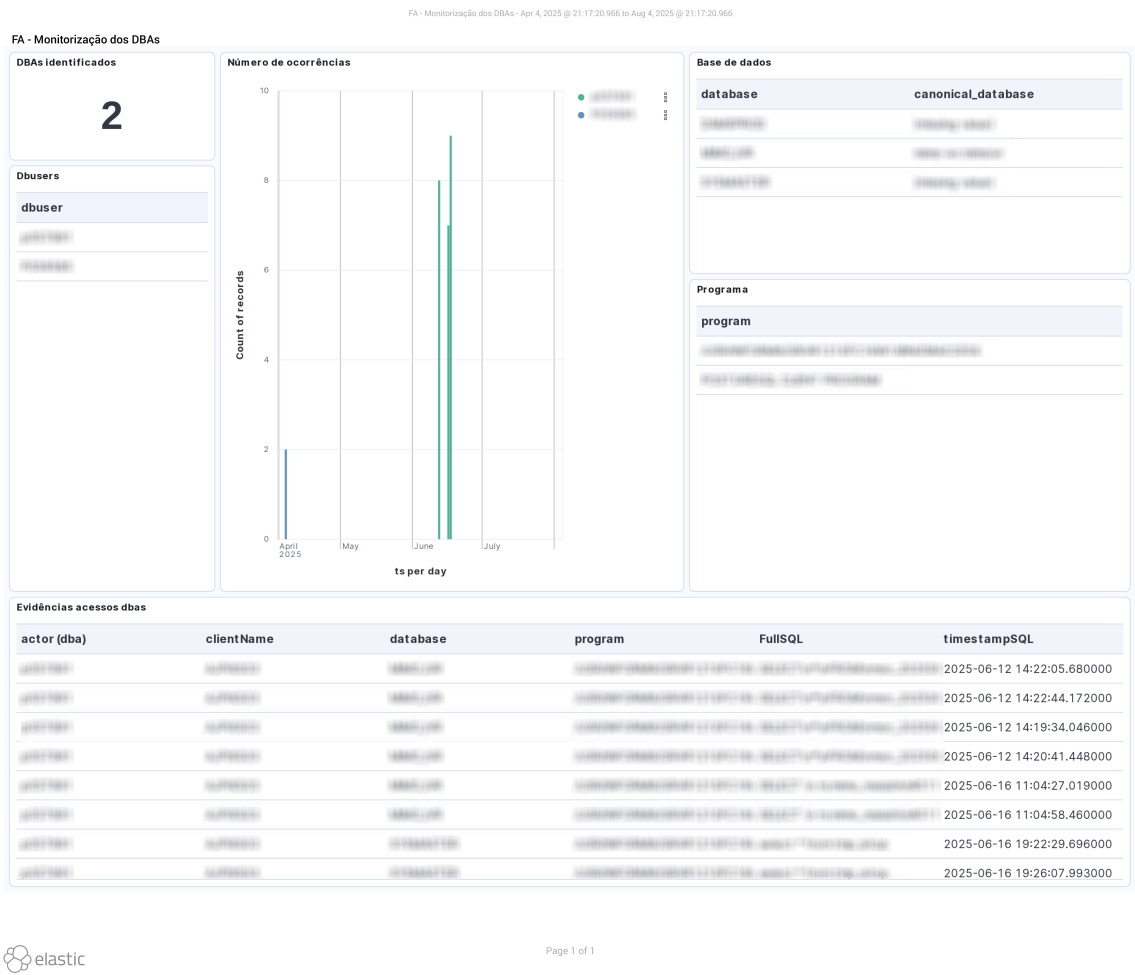


Figura 5.15: *Dashboard* da monitorização de acessos dos DBAs a dados críticos. O mesmo contém métricas como o número de casos detetados, evidências dos mesmos e um gráfico temporal que mostra a evolução temporal dos casos detetados.

Estes elementos permitem uma gestão mais eficaz dos incidentes de segurança, facilitam auditorias e promovem uma cultura de responsabilidade e transparência no acesso a dados sensíveis.

Conclusão

A implementação do processo de monitorização de acessos por parte de DBAs a dados críticos revelou-se uma medida eficaz de reforço da segurança da informação. O sistema permite identificar, de forma automatizada, potenciais acessos indevidos, garantindo capacidade de resposta rápida.

Os testes realizados demonstraram que a solução é funcional e sensível a cenários reais, como comprovado pelo caso detetado e posteriormente validado.

Recomenda-se a continuidade deste processo com revisões periódicas das políticas de monitorização, bem como o reforço da colaboração entre as equipas de segurança e os DBAs. Esta abordagem preventiva e colaborativa contribui para um ambiente mais seguro, eficiente e alinhado com os princípios de governança e conformidade organizacional.

Capítulo 6

Conclusão e Trabalho Futuro

6.1 Conclusão

O propósito deste projeto foi implementar mecanismos de segurança que permitissem identificar acessos suspeitos de utilizadores nominais em bases de dados, através de um processo totalmente automatizado, desde a fase de deteção, *reporting* e criação de métricas. Tal como descrito no [Capítulo 5](#), foram definidos, implementados e testados três casos de uso distintos: *HouseKeeping BDs*, Partilha de Contas e Monitorização de DBAs, suportados por tecnologias como o *IBM Guardium* e o *Elastic Stack*, bem como por *scripts* desenvolvidos em *Python*.

A abordagem seguida incluiu a definição de casos de uso representativos, a recolha e normalização de dados de diferentes fontes e a implementação de processos automatizados de reporte e de *dashboards*, que permitem monitorizar em tempo quase real situações anómalas e disponibilizar evidências claras para análise e tomada de decisão.

Durante o desenvolvimento, surgiram desafios relevantes, entre os quais a complexidade do ambiente de fontes de dados, a diversidade de fontes de informação e o volume de dados. Estes obstáculos foram superados através da iteração contínua e do ajuste progressivo das regras, critérios de deteção e constante alteração e aprendizagem das *queries* de SQL.

Apesar das limitações identificadas, os resultados confirmaram a eficácia da solução. Os casos de uso implementados permitiram detetar anomalias reais, nomeadamente acessos a contas de colaboradores já saídos da organização, situações de partilha de contas e acessos a dados críticos por DBAs realizados por DBAs. Na partilha de contas, registou-se apenas um número muito reduzido de falsos positivos, o que reforça a fiabilidade do método adotado. Adicionalmente, a disponibilização de relatórios e *dashboards* aumentou significativamente a visibilidade sobre os riscos e apoiou uma resposta mais rápida e fundamentada a potenciais incidentes.

Importa também salientar que todo o projeto foi realizado de acordo com o cronograma inicialmente definido no diagrama de Gantt [A.1](#), cumprindo as metas e prazos estabelecidos desde a fase de planeamento até à implementação final.

6.2 Trabalho Futuro

Embora os resultados obtidos tenham demonstrado a eficácia da solução implementada, existem diversas oportunidades de evolução que poderão aumentar a sua robustez, abrangência e utilidade prática. O trabalho futuro deverá incidir sobre melhorias técnicas, expansão de cenários e maior automatização dos processos de monitorização. Entre as principais direções de desenvolvimento destacam-se:

- **Expandir a monitorização da partilha de contas.** O conjunto de cenários considerados pode ser alargado, de forma a maximizar a identificação de casos de partilha de conta. Podem também ser incluído situações excecionais que podem ser confundidas com potenciais partilhas de contas, como a utilização pontual de um computador fixo no escritório quando um colaborador se esquece do portátil. Estas exceções deverão ser tratadas para reduzir ainda mais falsos positivos.
- **Enriquecimento das fontes de dados.** Poderá ser criada uma fonte de dados consolidada que contenha informação agregada de várias fontes. Dessa forma este e outros projetos podem consultar uma fonte de dados consolidada de forma mais simples e eficiente.
- **Aumentar a precisão no cálculo de severidade.** As fórmulas atuais poderão ser refinadas, incorporando mais fatores que permitam avaliar melhor a gravidade de cada situação. No caso da partilha de contas, por exemplo, poderão ser considerados elementos adicionais, como o horário do acesso (dentro ou fora do expediente), a relação hierárquica ou departamental dos utilizadores envolvidos, ou ainda o tipo de *software* utilizado no acesso.
- **Automatizar o *reporting* da Monitorização de DBAs.** Atualmente, o *reporting* deste caso de uso tem uma parte manual, visto que são feitas validações de forma a identificar possíveis falsos positivos antes de reportar a situação. Dada a criticidade deste caso de uso, atingindo um nível de maturidade superior, poderá ser automatizado o reporte de situações anómalas identificadas.
- **Expandir o número de casos de uso monitorizados.** No momento da redação deste projeto, foram tratados apenas três casos de uso. Aumentar este número permitirá ampliar a cobertura e reforçar a deteção de acessos anómalos de utilizadores de BDs em diferentes contextos.
- **Enriquecimento ETL.** Garantir que os logs de acessos via VPN sejam correlacionados com os logs de atribuição de IP, permitindo a associação automática entre cada IP e o utilizador a quem foi atribuído. Dessa forma, torna-se possível identificar cenários de partilha de contas, por exemplo, quando a conta da pessoa A aparece associada ao mesmo IP atribuído à pessoa B.
- **Maximizar abrangência do caso de uso *HouseKeeping BD*.** Existem situações excecionais em que determinados utilizadores não são identificados pelo HK BD por não consta-

rem em nenhuma das fontes de dados analisadas. O mesmo acontece porque o utilizador não possui a nomenclatura padrão devido ao uso do mesmo em tecnologias antigas. Para colmatar estas lacunas, pode ser implementada no caso de uso uma lógica de associação entre identidade e conta, permitindo assim obter informação atualizada sobre o estado do utilizador na empresa.

- **Aumentar âmbito de BD.** No projeto foram consideradas apenas algumas BDs de produção. No entanto, os casos de uso podem ser expandidos para abranger outras BDs críticas.

Bibliografia

- [1] 205: Policy management. Disponível em: <https://vest.buildlab.cloud/en/guardium/level-4/205> [Acedido em: 01-07-2025].
- [2] Using session-level and advanced session-level policies - ibm documentation, 3 2025. Disponível em: <https://www.ibm.com/docs/en/gdp/11.5.0?topic=policies-using-session-level-advanced-session-level> [Acedido em: 01-07-2025].
- [3] José Águas. Detecção de acessos anómalos a bases de dados críticas. Tese de mestrado em engenharia informática, Universidade de Lisboa (Portugal), 2020. Disponível em: <https://repositorio.ul.pt/handle/10451/48349> [Acedido em: 13-11-2024].
- [4] David Brimley. What is an elasticsearch index?, 2023. Disponível em: <https://www.elastic.co/pt/blog/what-is-an-elasticsearch-index/> [Acedido em: 26-11-2024].
- [5] CISA. Defining insider threats, 2024. Disponível em: <https://www.cisa.gov/topics/physical-security/insider-threat-mitigation/defining-insider-threats> [Acedido em: 13-11-2024].
- [6] CNPD. Cnpd. Disponível em: <https://www.cnpd.pt/cnpd/o-que-somos-e-quem-somos/> [Acedido em: 26-11-2024].
- [7] ENISA. Definição insider threat cncs, 2018. Disponível em: <https://www.cncs.gov.pt/pt/glossario/> [Acedido em: 28-11-2024].
- [8] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). <https://data.europa.eu/eli/reg/2016/679/oj>, May 2016. Disponível em: <https://data.europa.eu/eli/reg/2016/679/oj> [Acedido em: 26-11-2024].
- [9] Zilin Huang, Xiangyan Tang, Hongyu Li, Xinyi Cao, and Jieren Cheng. Tabsec: A collaborative framework for novel insider threat detection, 2024. Disponível em: <https://arxiv.org/abs/2411.01779> [Acedido em: 21-09-2025].

- [10] IBM. O que são ameaças internas?, 2024. Disponível em: <https://www.ibm.com/br-pt/topics/insider-threats> [Acedido em: 13-11-2024].
- [11] Justice.gov. Former employee of medical packaging company charged with sabotaging electronic shipping records leading to the delay of ppe to healthcare providers, 2020. Disponível em: <https://www.justice.gov/usao-ndga/pr/former-employee-medical-packaging-company-allegedly-sabotages-electronic-shipping> [Acedido em: 13-11-2024].
- [12] Justice.gov. Former twitter employee found guilty of acting as an agent of a foreign government and unlawfully sharing twitter user information, 2022. Disponível em: <https://www.justice.gov/opa/pr/former-twitter-employee-found-guilty-acting-agent-foreign-government-and-unlawfully-sharing> [Acedido em: 13-11-2024].
- [13] Koch, Raúl. Descoberta e classificação automática de informação crítica em bases de dados. Tese de mestrado em ciência de dados, Universidade de Lisboa (Portugal), 2022. Disponível em: <http://hdl.handle.net/10451/59015> [Acedido em: 13-11-2024].
- [14] Lusa. Hospital do barreiro contesta judicialmente coima de 400 mil euros de comissão de dados, 2018. Disponível em: <https://www.publico.pt/2018/10/22/sociedade/noticia/hospital-barreiro-contesta-judicialmente-coima-400-mil-euros-comissao-dados-1848479> [Acedido em: 26-11-2024].
- [15] Hanna Mazzawi, Gal Dalal, David Rozenblatz, Liat Ein-Dorx, Matan Niniox, and Ofer Lavi. Anomaly detection in large databases using behavioral patterning. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1140–1149, 2017. Disponível em: <https://doi.org/10.1109/ICDE.2017.158> [Acedido em: 13-11-2024].
- [16] Microsoft. 4725(s): A user account was disabled, 2021. Disponível em: <https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4725> [Acedido em: 22-03-2025].
- [17] Ivo Pascoal. Poirotdbv2: deteção, qualificação, avaliação e resposta rápida a incidentes de acesso ilícito ou exfiltração ilícita de informação crítica. Tese de mestrado em ciência de dados, Universidade de Lisboa (Portugal), 2021. Disponível em: <https://repositorio.ul.pt/handle/10451/51901> [Acedido em: 13-11-2024].
- [18] Altice Portugal. Política de segurança da informação do grupo altice portugal, 2024. Disponível em: <https://conteudos.altice.pt/institucional/documentos/pt/seguranca/politica-seguranca-informacao.pdf> [Acedido em: 13-11-2024].

- [19] Zack Whittaker. Robinhood says millions of customer names and email addresses taken in data breach, 2021. Disponível em: <https://techcrunch.com/2021/11/09/robinhood-data-breach/> [Acedido em: 13-11-2024].
- [20] Olivia Wisbey. Choosing between a rule-based vs. machine learning system, 2024. Disponível em: <https://www.techtarget.com/searchenterpriseai/feature/How-to-choose-between-a-rules-based-vs-machine-learning-system> [Acedido em: 30-11-2024].

Apêndice A

Diagrama de Gantt



Figura A.1: Apresentação do diagrama de Gantt cumprido ao longo da tese.