INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Security Command HUB - Automação da Gestão de Vulnerabilidades e Planeamento de Testes de Segurança Integrados com a CMDB da Euronext

ANDRÉ FILIPE DA SILVA DUARTE
Junho de 2026

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Security Command HUB
Automação da Gestão de Vulnerabilidades e Planeamento de Testes de Segurança Integrados com a CMDB da Euronext

André Filipe da Silva Duarte

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Engenharia de Software

Orientador: Luís Nogueira

Porto, Junho 2026

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 27 de junho de 2026

Dedicatória

Quero dedicar este trabalho, em primeiro lugar, à minha noiva, pelo amor, pela paciência e pelo apoio constante ao longo de todo este percurso. Nos momentos de maior exigência, dúvida ou cansaço, foi sempre a sua presença e confiança que me permitiram continuar, especialmente nos momentos em que tudo pareceu mais difícil.

Aos meus pais, a quem devo muito, mas mesmo muito mais do que palavras, deixo o meu mais profundo obrigado. Sempre acreditaram em mim, deram tudo de si para que eu pudesse estudar e ser alguém na vida, e ensinaram-me o valor do esforço, da responsabilidade e do respeito.

Aos meus familiares mais próximos – avós, padrinhos e primos – agradeço o carinho, o incentivo e o apoio contínuo ao longo deste percurso académico e pessoal. A vossa presença foi fundamental para nunca me sentisse sozinho neste caminho.

Por fim, aos meus amigos mais próximos, que estiveram sempre presentes, agradeço não só a ajuda, mas também os momentos de descontração e amizade, que me permitiram desligar e manter a sanidade necessária para conciliar o trabalho, os estudos e a vida pessoal.

Resumo

Atualmente, organizações de grande dimensão gerem múltiplos ativos tecnológicos, como aplicações, instâncias e bases de dados, através de *Configuration Management Databases* (*CMDBs*) corporativas, como o *Jira Service Management*.

A equipa de segurança da Euronext é responsável pela validação e proteção desses ativos, assegurando a execução periódica de testes conforme a criticidade definida para cada um. Contudo, os processos de planeamento de testes de segurança e de atualização de inventário eram ainda conduzidos manualmente, recorrendo a folhas *Excel*, o que originava erros, atrasos e perda de rastreabilidade.

Em simultâneo, a informação sobre vulnerabilidades encontrava-se dispersa por várias fontes, exigindo agregação e qualidade de dados para suportar decisões de mitigação eficazes, conforme estabelecido no *Digital Operational Resilience Act* (DORA).

O problema central consistia na automação e orquestração da recolha de inteligência de vulnerabilidades e o planeamento de avaliações de segurança de forma integrada com a *CMDB*, garantindo precisão, escala e redução de esforço manual.

Neste contexto, foi concebido e implementado o ***Security Command Hub***, um sistema composto por dois módulos complementares: o *Security Automation Service* (*SAS*), responsável pela integração e enriquecimento de vulnerabilidades provenientes de múltiplas fontes, e o *Resilience Testing Framework* (*RTF*), orientado à gestão e planeamento automatizado de ativos e avaliações de segurança.

A solução adotou uma arquitetura moderna e escalável, baseada em microserviços, contentores e pipelines de CI/CD, com integração nativa com o *Jira Service Management*.

A abordagem foi validada através de uma prova de conceito em ambiente controlado, em colaboração com a equipa de segurança da Euronext. A avaliação experimental demonstrou melhorias estruturais no fluxo operacional, na qualidade dos dados consolidados, na coerência do planeamento automático e na rastreabilidade entre ativos, vulnerabilidades e *assessments*, contribuindo para o cumprimento dos requisitos de conformidade definidos pelo *DORA*.

Palavras-chave: Segurança, vulnerabilidades, planeamento, CMDB, automação, gestão de serviços, Euronext

Abstract

Large organizations currently manage multiple technological assets — such as applications, instances, and databases — through corporate *Configuration Management Databases* (CMDBs), such as Jira Service Management.

The Euronext Security Team is responsible for validating and protecting these assets, ensuring the periodic execution of security tests according to each asset's defined criticality. However, security test planning and inventory update processes were still performed manually, using spreadsheets, which led to errors, delays, and lack of traceability.

At the same time, vulnerability information was scattered across several sources, requiring data aggregation and quality assurance to support effective mitigation decisions, as outlined in the Digital Operational Resilience Act (DORA).

The core problem consisted of automating and orchestrating the collection of vulnerability intelligence and the planning of security assessments, fully integrated with the CMDB, ensuring accuracy, scalability, and reduced manual effort.

To address this problem, the Security Command Hub was designed and implemented – a system composed of two complementary modules: the Security Automation Service (SAS), responsible for vulnerability integration and enrichment from multiple sources, and the Resilience Testing Framework (RTF), focused on automated asset management and security assessment planning.

The solution adopted a modern and scalable architecture, based on microservices, containers, and CI/CD pipelines, with native integration with Jira Service Management.

The proposed approach was validated through a proof of concept in a controlled environment, in collaboration with the Euronext Security Team. The experimental evaluation demonstrated structural improvements in the operational workflow, in the quality of consolidated vulnerability data, in the consistency of automated planning, and in the traceability between assets, vulnerabilities, and assessments – contributing to the compliance requirements defined by DORA.

Keywords: Security, vulnerability, planning, CMDB, automation, service management, Euronext

Agradecimentos

Gostaria de expressar o meu sincero agradecimento ao meu orientador, Luís Nogueira, pelo acompanhamento, disponibilidade e contributo crítico ao longo de todo o desenvolvimento deste trabalho. Os seus conselhos, comentários construtivos e exigência foram fundamentais para a evolução do projeto e para o rigor científico existente neste documento.

Agradeço igualmente à Euronext pela oportunidade de desenvolver este projeto em contexto real, bem como pela confiança, abertura e colaboração demonstradas ao longo de todo o processo. O contacto direto com um ambiente profissional exigente foi determinante para a relevância prática deste trabalho e para o meu crescimento enquanto profissional.

Ao Instituto Superior de Engenharia do Porto, agradeço a formação académica sólida e exigente pela qual é reconhecido e que também me foi proporcionada. Juntamente com o enquadramento científico, tornou possível a realização deste projeto.

Por fim, deixo um agradecimento especial a todos os colegas com quem tive o privilégio de trabalhar ao longo do meu percurso académico e profissional. Toda a partilha de conhecimento, o espírito de equipa e os desafios enfrentados em conjunto, contribuíram de forma decisiva para a minha evolução, profissional e pessoal, ajudando-me a ser a pessoa que sou hoje.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

DORA	*Digital Operational Resilience Act*
CMDB	*Configuration Management Database*
NVD	*National Vulnerability Database*
NIST	*National Institute of Standards and Technology*
CVE	*Common Vulnerabilities and Exposures*
CVSS	*Common Vulnerability Scoring System*
SAS	*Security Automation Service*
RTF	*Resilience Testing Framework*
API	*Application Programming Interface*
RGPD	Regulamento Geral sobre a Proteção de Dados
ITSM	*IT Service Management*
TIC	Tecnologias de Informação e Comunicação
ESMA	*European Securities and Markets Authority*
ITIL	*Information Technology Infrastructure Library*
SIEM	*Security Information and Events Management*
SSO	*Single Sign-On*

1 Introdução

Este capítulo descreve o contexto do projeto, o problema identificado, os objetivos atingidos e consequentemente os resultados obtidos. Complementarmente, são apresentadas as metodologias adotadas e a forma como o documento está estruturado.

1.1 Contexto

A segurança da informação tornou-se uma prioridade nas organizações que gerem grandes volumes de ativos tecnológicos, principalmente em contextos críticos e regulados, onde a proteção da informação e a continuidade operacional são essenciais (ISO/IEC 27001:2022, 2022; ENISA, 2023).

A crescente exigência regulatória na área da cibersegurança, em especial com a entrada em vigor do *Digital Operational Resilience Act*[1] *(DORA)*, reforça a necessidade de as organizações financeiras garantirem a rastreabilidade, automatização e fiabilidade dos seus processos de segurança operacional (Scott, 2021; European Union, 2022; European Commission, 2024).

Neste enquadramento, a Euronext procurava evoluir de processos manuais para uma abordagem automatizada, integrando a gestão de ativos e vulnerabilidades numa solução centralizada e auditável.

A Euronext, como operador europeu de mercados financeiros, mantém milhares de aplicações, instâncias e bases de dados registadas na sua *CMDB*[2] corporativa, suportada pelo *Jira Service*

[1] Digital Operational Resilience Act – é uma regulamentação da União Europeia que visa reforçar a cibersegurança no setor financeiro.
[2] *Configuration Management Database* – é uma base de dados de gestão que esclarece as relações entre o hardware, o software e as redes utilizadas por uma organização de TI.

Management[3], que atua como repositório central da informação sobre ativos e respetivas relações (Atlassian, 2024; Alén, 2020).

A equipa de segurança é responsável por garantir a atualização e proteção contínua desses ativos, realizando testes periódicos que avaliam a sua exposição a vulnerabilidades e o cumprimento das políticas internas de conformidade e gestão de risco, alinhados sempre com as boas práticas de gestão de serviços de Tecnologias da Informação (TI), como as definidas pelo *ITIL* (Axelos, 2020; Agutter, 2020).

O crescimento do número de sistemas e a complexidade da infraestrutura criaram a necessidade de automatizar tarefas que ainda dependiam de folhas *Excel* e procedimentos manuais, os quais são reconhecidos na literatura como fontes frequentes de erro, inconsistência e baixa escalabilidade (Souppaya & Scarfone, 2022).

A transformação digital e o aumento de ameaças exigem soluções integradas que centralizem a informação, reduzam erros e acelerem a resposta a vulnerabilidades.

Neste contexto surgiu o projeto ***Security Command Hub***, desenvolvido em parceria com a Euronext, para criar uma plataforma que integra a gestão de ativos e a análise de vulnerabilidades de forma automatizada e auditável.

1.2 Problema

Atualmente, as organizações de grande dimensão gerem múltiplos ativos tecnológicos, como aplicações, instâncias e bases de dados, em CMDBs corporativas como elemento central de suporte à gestão de serviços e à *governance* de ativos de TI (Agutter, 2020; Alén, 2020). Em ambientes corporativos complexos, estas bases de dados são fundamentais para assegurar consistência, rastreabilidade e alinhamento entre ativos e processos.

Na Euronext, a equipa de segurança é responsável por garantir a proteção e conformidade desses ativos, realizando testes de segurança de forma periódica, de acordo com a criticidade e o impacto operacional de cada sistema, em alinhamento com as boas práticas de gestão de serviços e segurança de informação (Axelos, 2020; ISO/IEC 27001:2022, 2022).

Contudo, os processos de planeamento de testes de segurança e de atualização de inventário eram ainda conduzidos manualmente, recorrendo a folhas de *Excel*, o que resultava em erros, atrasos e fraca rastreabilidade.

Em simultâneo, a informação sobre vulnerabilidades encontrava-se dispersa por diversas fontes, como o *AWS Security Hub*, *Microsoft Defender for Cloud*, *Red Hat Insights*, *Security Scorecard* e *National Vulnerability Database (NVD)*, o que exigia consolidação, normalização e validação de

[3] Jira Service Management – é uma plataforma da Atlassian para gerenciamento de serviços, voltada para equipes de TI, desenvolvimento e negócios.

dados para uma análise eficaz e consistente do risco associado a cada ativo (Chatterjee, 2021; Diogenes & Janetscheck, 2022; Manne, 2024; Sohval, 2020).

Estas dificuldades tornavam o processo de gestão de vulnerabilidades ineficiente, dificultando a priorização de ações de mitigação e o cumprimento de políticas internas e normativas externas, como evidenciado por relatórios recentes sobre ameaças e maturidade em cibersegurança (ENISA, 2023).

O *DORA*, em vigor na União Europeia desde janeiro de 2025, reforça a necessidade de automatizar processos de segurança operacional, garantir rastreabilidade e assegurar a resiliência digital das instituições financeiras (European Union, 2022).

Neste contexto, o problema central consistia em automatizar e orquestrar a recolha de inteligência de vulnerabilidades e o planeamento de avaliações de segurança de forma integrada com a CMDB, assegurando precisão, escalabilidade e redução do esforço manual.

O projeto foi desenvolvido para permitir a integração entre sistemas, a consolidação de dados e a priorização automática de vulnerabilidades e testes, de modo a otimizar a resposta a riscos e promover a conformidade com os requisitos do *DORA*.

De ressalvar que, embora este trabalho seja conduzido como um caso de estudo na Euronext, o problema é representativo de um padrão recorrente em organizações de grande dimensão e reguladas, onde as evidências de vulnerabilidades encontram-se distribuídas por múltiplas plataformas, enquanto o inventário reside numa *CMDB* e a execução é gerida por sistema de *ticketing* (como o *Jira*). A ausência de integração e de regras de planeamento executáveis conduz a processos manuais, inconsistências de dados e dificuldades em produzir evidências auditáveis.

1.3 Objetivos

O objetivo geral do projeto consistiu em conceber, implementar e avaliar o *Security Command Hub*, uma plataforma que integra e automatiza a gestão de ativos e a análise de vulnerabilidades, assegurando sincronização contínua com a CMDB suportada pelo *Jira Service Management* (Atlassian, 2024; Alén, 2020).

A solução divide-se em dois módulos complementares. O primeiro, designado *Security Automation Service (SAS)*, centraliza a recolha de informação de múltiplas fontes de vulnerabilidades, como *AWS Security Hub*, *Red Hat Insights*, *Microsoft Defender*, *Security Scorecard* e NVD. Este módulo aplica processos de normalização e enriquecimento de dados, recorrendo a métricas de risco amplamente adotadas na literatura e na prática industrial, como

Common Vulnerability Scoring System (CVSS)[4], de forma a suportar uma avaliação consistente da severidade das vulnerabilidades (Walkowski et al., 2021; Souppaya & Scarfone, 2022).

O segundo, o *Resilience Testing Framework (RTF)*, foi desenvolvido para substituir os processos manuais de planeamento de testes de segurança. Este módulo permite automatizar a criação de planos de avaliação, a definição de prioridades e a atualização de tarefas e estados no Jira Service Management, promovendo maior consistência, rastreabilidade e alinhamento com boas práticas de gestão de serviços de TI e segurança da informação (Agutter, 2020; Axelos, 2020).

O projeto garantiu ainda uma integração programática com a *CMDB*, permitindo a sincronização bidirecional de ativos e *issues*, reduzindo discrepâncias entre sistemas e melhorando a qualidade da informação disponível para apoio à decisão. Foram desenvolvidos conectores estáveis para as diferentes fontes de segurança e aplicadas regras automáticas de priorização e agendamento baseadas em critérios de criticidade, exposição e impacto operacional, conforme recomendado por abordagens modernas de gestão de vulnerabilidades e automação de segurança (Kovacevic & DiCola, 2023; Rajapakse et al., 2022).

Por fim, o protótipo foi validado empiricamente através de cenários de teste controlados, observação direta do comportamento dos componentes e análise estrutural da arquitetura implementada. A avaliação incidiu sobre propriedades como consolidação multi-fonte, coerência de planeamento, rastreabilidade operacional e estabilidade técnica sob carga controlada, permitindo demonstrar viabilidade técnica e correção funcional da abordagem proposta, embora sem medir de forma conclusiva o seu impacto longitudinal em produção.

1.3.1 Questões de Investigação

Com base no problema identificado, nos objetivos definidos e na análise do estado da arte, este trabalho foi orientado por um conjunto de questões de investigação que visam avaliar empiricamente o impacto da automação e da integração de processos de segurança, em ambientes corporativos e regulados.

A questão de investigação principal é a seguinte:

- QI1 – Em que medida a automação do planeamento de testes de segurança e da deteção de vulnerabilidades, integrada com a *CMDB*, melhora a rastreabilidade operacional e a capacidade de demonstração de conformidade regulatória, quando comparada com processos manuais numa organização financeira como a Euronext?

Para suportar esta questão central, foram definidas as seguintes questões de investigação secundárias:

[4] Common Vulnerability Scoring System - é um padrão internacional usado para medir a gravidade de vulnerabilidades em sistemas de software e hardware.

- QI2 – Em que medida a consolidação de dados de vulnerabilidades provenientes de múltiplas fontes heterogéneas melhora a qualidade do *dataset* de vulnerabilidades, em termos de duplicação, completude e associação a ativos da *CMDB*?

- QI3 – Em que medida a normalização e a deduplicação de vulnerabilidades, recorrendo a métricas como o *CVSS*, reduzem a variabilidade e inconsistência na avaliação do risco, quando comparadas com abordagens manuais?

- QI4 – Em que medida a automação do cálculo de periodicidade e do agendamento de testes, baseada na criticidade dos ativos, produz planos de avaliação mais consistentes e previsíveis, quando comparados com o planeamento manual?

- QI5 – Em que medida a sincronização bidirecional com a *CMDB* reduz inconsistências de informação entre inventário, vulnerabilidades e processos de mitigação?

- QI6 – Em que medida a agregação centralizada de informação e evidência operacional reduz o esforço de preparação de evidência e melhora a rastreabilidade exigida pelo *DORA*?

No subcapítulo seguinte, é descrita a estratégia de avaliação adotada para responder a cada uma destas questões de investigação.

1.4 Abordagem

A abordagem do projeto seguiu uma estratégia incremental, priorizando a automação dos processos críticos de segurança e a integração contínua entre sistemas.

Numa fase inicial, foram estudadas as interações existentes entre as ferramentas utilizadas pela equipa de segurança da Euronext, com o objetivo de compreender dependências, fluxos de informação e limitações operacionais. Esta análise permitiu caracterizar o estado atual (*as-is*) dos processos de deteção de vulnerabilidades e planeamento de testes, identificando oportunidades de melhoria sustentadas por automação e integração (Souppaya & Scarfone, 2022; ENISA, 2023).

Com base nessa análise, foi desenhada uma arquitetura modular composta por dois módulos complementares: o *Security Automation Service (SAS)*, responsável pela agregação e enriquecimento de vulnerabilidades, e o *Resilience Testing Framework (RTF)*, responsável pelo planeamento automatizado de testes de segurança. Esta separação funcional segue princípios de modularidade e desacoplamento, amplamente defendidos na literatura sobre arquiteturas modernas e práticas DevOps e DevSecOps (Bass et al., 2015; Rajapakse et al., 2022).

A integração com a CMDB do *Jira Service Management* foi desenvolvida por meio de *Application Programming Interface* (APIs) que permitem sincronização bidirecional de ativos,

vulnerabilidades e estados de teste, bem como a atualização programática de *issues* associados aos processos de segurança (Atlassian, 2024; Alén, 2020). Esta abordagem visou reduzir inconsistências entre sistemas e assegurar que a informação operacional permanece atualizada e rastreável.

Esta abordagem procurou assegurar coerência técnica e eficiência operacional, assegurando alinhamento com as exigências do *DORA* e com os objetivos de resiliência digital da organização.

1.4.1 Estratégia de Avaliação e Métricas

Para além da abordagem técnica descrita, este trabalho definiu uma estratégia de avaliação orientada a responder às questões de investigação formuladas. Esta estratégia teve como objetivo garantir que a fase de implementação e validação do protótipo fosse orientada por critérios de avaliação claros, mensuráveis e alinhados com o problema identificado.

A avaliação foi baseada numa comparação entre o processo manual atualmente utilizado pela equipa de segurança da Euronext e a abordagem automatizada implementada pelo *Security Command Hub*. O processo manual, suportado por folhas *Excel* e consultas manuais a múltiplas ferramentas de segurança, foi utilizado como *baseline* para a análise comparativa.

Importa referir que a avaliação experimental foi conduzida sobre um subconjunto controlado de ativos. Esta delimitação resultou de restrições associadas ao contexto organizacional da solução, nomeadamente confidencialidade da informação, direitos sobre a solução desenvolvida e sensibilidade dos ativos envolvidos. Por essa razão, a avaliação não foi concebida como estudo estatístico em ambiente produtivo, mas como validação de um protótipo funcional em cenários controlados.

Neste enquadramento, a avaliação recorreu à observação direta do comportamento dos componentes, à análise de registos persistidos, *tickets* gerados no *Jira*, execução de cenários controlados e observação de métricas técnicas relacionadas com a arquitetura, incluindo testes de carga limitados sobre as *APIs* desenvolvidas. Esta opção metodológica permitiu validar propriedades técnicas e funcionais centrais da solução, mas não permitiu medir de forma conclusiva o seu impacto quantitativo em produção nem a sua estabilidade ao longo de ciclos operacionais prolongados.

As dimensões consideradas incluíram eficiência operacional, qualidade dos dados, qualidade do planeamento, rastreabilidade e evidência relevante para o *DORA*, bem como condições arquiteturais e desempenho sob carga controlada.

A recolha de evidência para avaliação foi efetuada através da análise de respostas produzidas pelo *SAS*, registos persistidos no *RTF*, *tickets* e atualizações no *Jira*, execução de cenários controlados e observação de métricas relacionadas com a arquitetura, assim como alguns testes de carga sobre as *APIs* produzidas.

A Tabela 1 apresenta a definição das dimensões de avaliação, do tipo de evidência utilizada e dos resultados esperados pelo protótipo ao longo da avaliação, garantindo que a fase de desenvolvimento foi acompanhada por um plano de avaliação sólido e consistente.

Tabela 1 – Mapeamento de Avaliação

Questão de investigação	Dimensão de avaliação	Tipo de evidência utilizada	Baseline de comparação	Resultado esperado no protótipo
QI1 – Impacto da automação na rastreabilidade e conformidade	Rastreabilidade e evidência relevante para o *DORA*	Cenários controlados, *tickets* no *Jira*, relações entre ativo, vulnerabilidades e *assessments*	Processo manual baseado em *Excel* e *Jira*	Demonstração de ligação explícita entre inventário, vulnerabilidades, *assessments* e evidência operacional
QI2 – Consolidação de vulnerabilidades multi-fonte	Qualidade dos dados	Respostas do *SAS* em cenários com múltiplas fontes e ativos associados	Consolidação manual via *Excel*	Demonstração de consolidação multi-fonte e associação coerente ao ativo
QI3 – Normalização e deduplicação (*CVSS*)	Qualidade dos dados	Cenários controlados com resultados heterogéneos e vulnerabilidades repetidas	Avaliação manual por analistas	Demonstração de normalização da resposta e eliminação de duplicações no contexto do protótipo
QI4 – Planeamento automático de *assessments*	Qualidade do planeamento	Registos persistidos no *RTF* e execução dos cenários controlados	Planeamento manual em *Excel*	Demonstração da coerência funcional entre atributos do ativo, prioridade, tipo de teste e data sugerida
QI5 – Sincronização bidirecional com a *CMDB*	Integração e consistência do inventário	Observação da sincronização entre *CMDB*, *RTF* e *Jira* em cenários controlados	Atualizações manuais e periódicas	Demonstração de integração funcional com a *CMDB* e utilização do inventário como base do planeamento
QI6 – Evidência e rastreabilidade para o *DORA*	Rastreabilidade e evidência relevante para o *DORA*	Fluxos *end-to-end*, *tickets* gerados e histórico de *assessments*	Preparação manual de evidência	Demonstração de produção de evidência operacional mais estruturada e centralizada

Este alinhamento entre questões de investigação e estratégia de avaliação permitiu responder de forma rigorosa às propriedades que podiam ser observadas ao nível de um protótipo

funcional. Em contrapartida, as dimensões que exigem medição longitudinal, uso continuado em produção ou comparação quantitativa extensiva foram assumidas como parcialmente respondidas e discutidas criticamente no capítulo 5.

1.5 Contributos

O presente trabalho oferece contributos técnicos, metodológicos e organizacionais no domínio da automação da segurança da informação e da gestão de vulnerabilidades em ambientes regulados.

Os principais contributos técnicos e metodológicos deste trabalho são os seguintes:

1. A proposta e implementação de um modelo integrado que articula inventário de ativos mantido em *CMDB*, múltiplas fontes de vulnerabilidades e planeamento de *assessments* de segurança num fluxo coerente e rastreável.

2. O desenvolvimento de um mecanismo de consolidação, normalização e deduplicação de vulnerabilidades provenientes de fontes heterogéneas, orientado ao ativo e ao seu contexto operacional.

3. A definição e implementação de um conjunto de algoritmos de apoio ao planeamento, incluindo priorização de ativos, determinação do tipo de teste e sugestão de agendamento de *assessments* com base em criticidade, exposição, contexto regulamentar e histórico de avaliações.

4. A definição de uma abordagem de avaliação alinhada com as questões de investigação do trabalho, baseada em cenários controlados, critérios verificáveis e evidência observável sobre eficiência operacional, qualidade de dados, qualidade do planeamento, rastreabilidade e evidência relevante para o *DORA*.

Para além destes contributos, o trabalho também oferece contributos práticos e organizacionais.

5. A materialização da abordagem proposta sob a forma de um protótipo funcional, designado de *Security Command Hub*, composto por dois módulos desacoplados: o *Security Automation Service* (*SAS*) e o *Resilience Testing Framework* (*RTF*).

6. A definição de um modelo de integração que preserva as ferramentas de segurança já utilizadas pela organização, evitando substituição integral do ecossistema existente e permitindo evolução progressiva da solução.

7. A criação de um mecanismo de suporte à produção de evidência operacional e à rastreabilidade entre ativos, vulnerabilidades e *assessments*, com relevância direta para auditorias e para a conformidade regulamentar.

Embora a implementação realizada tenha sido motivada por um caso de estudo concreto, o principal contributo científico deste trabalho não reside na utilização de tecnologias específicas, mas na definição de uma abordagem integrada para a automatização da gestão de vulnerabilidades e do planeamento de *assessments* em ambientes regulados.

A articulação entre o inventário mantido em *CMDB*, consolidação multi-fonte de vulnerabilidades, mecanismos de priorização orientados ao contexto operacional e geração estruturada de evidência constitui uma proposta coerente que procura responder a lacunas identificadas na literatura e na prática industrial.

Neste sentido, o trabalho contribui não apenas com um protótipo funcional, mas também com um modelo arquitetural, um conjunto de mecanismos de integração e uma abordagem de avaliação que poderão servir de base a futuras implementações e investigações neste domínio.

1.6 Considerações Éticas

O projeto foi conduzido em conformidade com princípios éticos fundamentais e políticas internas da Euronext, assegurando confidencialidade, privacidade e integridade da informação processada ao longo de todas as fases do trabalho. Estas preocupações estão alinhadas com as boas práticas reconhecidas na área da engenharia de *software* e da segurança da informação, bem como com os códigos de ética profissional amplamente adotados na área das tecnologias de informação (Brinkman et al., 2017).

Todos os testes e simulações utilizaram dados anonimizados ou artificiais, evitando a exposição de sistemas ou vulnerabilidades reais. Desta forma, assegurou-se uma redução dos riscos operacionais e éticos, garantindo que o desenvolvimento e a avaliação da solução não interferiram com o funcionamento normal da infraestrutura da organização (ISO/IEC 27001:2022, 2022).

A recolha e utilização de dados seguiram o princípio da minimização, restringindo-se apenas à informação necessária para a validação das funcionalidades.

Durante o desenvolvimento, foram implementados controlos de acesso baseados em papéis e mecanismos de auditoria, assegurando conformidade com o **Regulamento Geral sobre a Proteção de Dados (RGPD)** (European Parliament & Council, 2016; Zaeem & Barber, 2020).

A condução do projeto incluiu a revisão prévia das permissões necessárias e a validação de segurança junto das equipas responsáveis, garantindo que nenhuma ação comprometeu os sistemas reais da organização.

1.7 Estrutura do documento

O presente documento organiza-se em capítulos que refletem a progressão natural do projeto, desde o enquadramento inicial até à análise técnica e científica das soluções propostas.

O Capítulo 1 – Introdução apresenta o contexto do projeto, o problema identificado, os objetivos definidos, a abordagem adotada, os contributos obtidos e as considerações éticas associadas.

O Capítulo 2 – Estado da Arte analisa o conhecimento existente sobre resiliência operacional digital, gestão de ativos, processos de deteção de vulnerabilidades e planeamento de avaliações de segurança, apresentando ainda os resultados da pesquisa que fundamentam o desenho do *Security Command Hub*.

O Capítulo 3 – Proposta de Solução e Design apresenta a solução proposta para responder ao problema identificado. São definidos os requisitos funcionais e não funcionais, descrita a arquitetura do *Security Command Hub*, os modelos de dados dos seus componentes principais, fluxos operacionais, algoritmos e as decisões arquiteturais principais.

O Capítulo 4 – Implementação descreve a implementação da solução proposta, apresentando a metodologia adotada, as tecnologias utilizadas e os principais mecanismos desenvolvidos, incluindo funcionalidades chave, processos automáticos e considerações ao nível da segurança, escalabilidade e *deployment*.

O Capítulo 5 – Avaliação Experimental apresenta a metodologia de avaliação adotada, o ambiente experimental e as métricas utilizadas para avaliar a solução. São depois apresentados os resultados obtidos, seguidos de uma análise crítica relativamente ao grau de resposta às questões de investigação formuladas.

O Capítulo 6 – Conclusões apresenta a síntese do trabalho realizado, discutindo as limitações identificadas, o impacto organizacional da solução e as ameaças à validade do estudo, bem como possíveis direções para trabalho futuro.

2 Estado da Arte

O presente capítulo tem como objetivo analisar o estado da arte relacionado com a gestão de vulnerabilidades, automação de processos de segurança e integração com a *CMDB*. Esta análise permitirá compreender o enquadramento normativo, técnico e metodológico que sustenta o desenvolvimento do *Security Command Hub*, servindo de base à definição da sua arquitetura e funcionalidades.

Na primeira secção, é apresentada a pesquisa exploratória, que aborda as principais referências regulamentares e técnicas, incluindo o *DORA* e o papel das *CMDBs* na gestão de ativos e alterações.

Seguidamente, são descritos os processos de deteção de vulnerabilidades e de planeamento de avaliações de segurança, fundamentais para o contexto do projeto.

A segunda secção apresenta os resultados da pesquisa, detalhando as lacunas identificadas, as conclusões extraídas sobre integração com *CMDBs* e descrevendo os dois módulos propostos: o *Security Automation Service (SAS)* e o *Resilience Testing Framework (RTF)*. Para cada módulo, é analisado o seu propósito, as integrações e, quando aplicável, os seus mecanismos de orquestração.

Por fim, é realizada uma análise crítica que sintetiza as observações recolhidas e identifica lacunas e oportunidades de inovação, justificando as opções de design e de implementação do protótipo.

Este enquadramento teórico e técnico assegura que o projeto se apoia em práticas consolidadas de gestão de serviços, ciber-resiliência e automação de segurança, promovendo alinhamento com as normas internacionais e com os requisitos do *DORA*.

Este capítulo não pretende apenas descrever normas, ferramentas e processos, porque a descrição isolada não explica decisões de *design*. O capítulo procura evidenciar limites e lacunas que impedem a automação de ponta a ponta, sobretudo quando o inventário reside numa *CMDB* e a evidência vive em múltiplas plataformas.

2.1 Pesquisa

A pesquisa bibliográfica e técnica realizada teve como objetivo identificar as principais referências normativas, metodológicas e tecnológicas relacionadas com a gestão de vulnerabilidades, automação de processos de segurança e integração com bases de dados de configuração (*CMDBs*).

O objetivo desta análise é estabelecer uma base sólida de conhecimento que sustente o desenho e a implementação do *Security Command Hub*, assegurando o alinhamento com as práticas internacionais e as exigências legais do setor financeiro.

A revisão de literatura incidiu sobre quatro domínios principais:

1. **Enquadramento regulatório europeu**, com destaque para o DORA, que define obrigações de ciber-resiliência para entidades financeiras.

2. **Sistemas de gestão de configuração (*CMDBs*)** e o seu papel na manutenção de inventários de ativos tecnológicos e nas operações de *IT Service Management (ITSM)*.

3. **Mecanismos de deteção e gestão de vulnerabilidades**, incluindo plataformas automatizadas e repositórios de informação de segurança.

4. **Processos de planeamento de avaliações de segurança**, que permitem avaliar periodicamente a exposição de sistemas críticos e o cumprimento das políticas internas.

A pesquisa confirma um padrão recorrente: referências normativas descrevem obrigações e controlos, mas não descrevem mecanismos técnicos para operacionalizar integração e rastreabilidade. Esta diferença entre "o que é exigido" e "como se executa" cria espaço para soluções orientadas à automação e à consistência de dados.

2.1.1 Metodologia de Revisão do Estado da Arte

A revisão do estado da arte teve como objetivo enquadrar cientificamente o problema abordado neste projeto, identificar abordagens existentes no domínio da gestão de vulnerabilidades e do planeamento de testes de segurança, assim como analisar lacunas técnicas e metodológicas que justifiquem o desenvolvimento do *Security Command Hub*. A metodologia adotada procurou garantir rigor académico, relevância prática e alinhamento com contextos organizacionais regulados, como o setor financeiro.

A pesquisa foi conduzida com base numa abordagem sistematizada, combinando fontes científicas, normas técnicas e documentação institucional reconhecida. Foram utilizadas bases de dados científicas aceites na área da engenharia informática e da segurança de informação, nomeadamente *IEEE Xplore* e *Google Scholar*, privilegiando artigos revistos por pares, livros técnicos e estudos de revisão. Em complemento, foram analisadas normas e publicações de

entidades reguladoras e organismos de referência, como a *European Union*, *ENISA*, *NIST* e *ISO*, particularmente relevantes no contexto da resiliência operacional digital e da conformidade regulatória.

A estratégia de pesquisa assentou na utilização de palavras-chave e combinações temáticas relacionadas com o domínio do projeto, nomeadamente: *vulnerability management*, *security automation*, *CMDB*, *ITIL*, *security assessments*, *DevSecOps*, *SOAR*, *risk-based prioritization* e *DORA*.

Foram definidos critérios de inclusão e exclusão das fontes analisadas. Como critérios de inclusão, consideraram-se trabalhos publicados maioritariamente nos últimos cinco a oito anos, com relevância científica comprovada, aplicabilidade a ambientes empresariais de média e grande dimensão e ligação direta aos temas da automação, integração de sistemas ou gestão de risco em segurança da informação. Foram também incluídas normas e regulamentos oficiais, independentemente da data de publicação, sempre que relevantes para o enquadramento legal e normativo do problema. Como critérios de exclusão, foram descartadas fontes excessivamente genéricas, conteúdos puramente comerciais, documentação sem validação técnica ou científica e artigos cuja abordagem não fosse aplicável a contextos organizacionais complexos ou regulados.

A análise das fontes selecionadas foi conduzida de forma temática e crítica, evitando uma numeração descritiva de ferramentas ou tecnologias. Os resultados foram organizados em domínios conceptuais – enquadramento regulatório, *CMDB* e *ITSM*, deteção de vulnerabilidades, planeamento de *assessments* e automação de processos de segurança – permitindo identificar convergências, limitações recorrentes e desafios não resolvidos na literatura e na prática industrial.

Esta abordagem metodológica permitiu não só sintetizar o conhecimento existente, mas também identificar lacunas relevantes, nomeadamente a ausência de soluções integradas que articulem *CMDB*, múltiplas fontes de vulnerabilidades e mecanismos automatizados de planeamento orientado ao risco. Essas lacunas fundamentam a proposta do *Security Command Hub* e orientam as questões de investigação e os objetivos do projeto.

2.1.2 Digital Operational Resilience Act (DORA)

O *DORA* é um regulamento europeu que estabelece um quadro harmonizado para a resiliência operacional digital no setor financeiro (European Union, 2022).

Aprovado em dezembro de 2022 e em vigor desde 17 de janeiro de 2025, o *DORA* complementa outras diretivas de segurança, como o *NIS2*[5] e o RGPD, introduzindo regras específicas para a

[5] NIS2 – É uma diretiva da União Europeia concebida para reforçar a cibersegurança em setores críticos e importantes, estabelecendo medidas obrigatórias de gestão de riscos, comunicação de incidentes e governação para as organizações abrangidas pelo seu âmbito de aplicação.

gestão de riscos de tecnologias de informação e comunicação (TIC) nas instituições financeiras europeias.

O principal objetivo do *DORA* é garantir que todas as entidades do setor financeiro sejam capazes de resistir, responder e recuperar de incidentes de cibersegurança, minimizando o impacto sobre a estabilidade económica e a confiança dos mercados.

Este regulamento abrange bancos, bolsas de valores, seguradoras, fundos de investimento, intermediários financeiros e fornecedores de serviços tecnológicos considerados críticos.

O *DORA* estrutura-se em cinco pilares fundamentais (European Commission, 2024):

1. **Gestão de risco TIC** – exige que as organizações estabeleçam um quadro de *governance* e controlo dos riscos de TI, incluindo políticas de segurança, continuidade e resposta a incidentes.

2. **Gestão e reporte de incidentes** – impõe a notificação obrigatória de incidentes de cibersegurança significativos às autoridades competentes, como a *European Securities and Markets Authority* (ESMA) ou o Banco Central Europeu.

3. **Testes de resiliência digital** – requer a realização regular de testes de penetração e avaliações de vulnerabilidades, executados por entidades qualificadas e supervisionados por autoridades competentes.

4. **Gestão de riscos de terceiros** – estabelece obrigações de *due diligence* e monitorização contínua de fornecedores de TIC, especialmente os classificados como *Critical ICT Third-Party Providers*.

5. **Partilha de informações e colaboração** – incentiva a cooperação entre entidades financeiras e autoridades na troca de informações sobre ameaças e boas práticas de mitigação.

O *DORA* está intimamente ligado à resiliência operacional e à gestão de vulnerabilidades, uma vez que obriga as instituições a manter inventários atualizados de ativos, registos de incidentes e planeamentos de testes periódicos. Este enquadramento legal torna essencial a existência de ferramentas automatizadas que suportem a recolha de dados, o agendamento de avaliações e o acompanhamento de resultados — precisamente as funcionalidades que o *Security Command Hub* pretende oferecer.

Ao promover a automação e a integração com sistemas de gestão de configuração, o *DORA* incentiva o desenvolvimento de plataformas centralizadas que consolidem dados de ativos, vulnerabilidades e riscos (ISO/IEC 27001:2022, 2022; ENISA, 2023).

Desta forma, regulações como o *DORA* têm sido catalisadoras da transformação digital na área da cibersegurança, incentivando o uso de arquiteturas escaláveis, pipelines de *CI/CD* e integrações nativas com *CMDBs*.

No entanto, apesar de o *DORA* definir metas claras de resiliência e evidência, não define um modelo único para inventário, correlação de vulnerabilidades e planeamento de testes. Esta ausência força cada organização a desenhar o seu próprio mecanismo de integração, o que aumenta o risco de heterogeneidade, *gaps* operacionais e evidência incompleta.

2.1.3 Configuration Management Database (CMDB)

A *CMDB* é um repositório central de informação sobre ativos de TI e sobre as relações que esses ativos mantêm entre si (Agutter, 2020; Axelos, 2020). O seu papel principal consiste em suportar uma representação estruturada do inventário tecnológico da organização, permitindo compreender dependências, avaliar impacto operacional e apoiar decisões sobre risco, mudança e continuidade de serviço.

No contexto da *ITIL* 4 (*Information Technology Infrastructure Library*), a *CMDB* integra o processo de *Service Asset and Configuration Management* e suporta o registo de *Configuration Items* (*CIs*), como aplicações, servidores, bases de dados, redes, contas técnicas e contratos de *software* (Axelos, 2020). Esta capacidade torna a *CMDB* particularmente relevante em ambientes complexos e regulados, onde a qualidade do inventário condiciona diretamente a rastreabilidade dos processos e a capacidade de auditoria.

A literatura associa a *CMDB* a benefícios como maior controlo do inventário, melhor visibilidade sobre relações entre ativos e melhor suporte à governação dos serviços de TI. Em contextos regulados, estes benefícios ganham importância adicional, porque a organização precisa de justificar prioridades, demonstrar cobertura de controlos e produzir evidência operacional coerente.

Na Euronext, a *CMDB* é suportada pelo *Jira Service Management*, através do módulo *Assets*, que permite representar ativos e relações de forma extensível (Atlassian, 2024). Cada ativo é modelado como objeto com atributos relevantes para operação e segurança, incluindo identificação, categoria, estado, criticidade, responsável e relações com outros elementos do ecossistema tecnológico.

O interesse desta abordagem, no contexto do presente trabalho, não reside apenas na capacidade de armazenar inventário. O ponto central está no facto de a *CMDB* fornecer o contexto necessário para interpretar vulnerabilidades e para suportar decisões de planeamento. Sem essa base, a informação técnica produzida por *scanners* e ferramentas de segurança permanece desligada da criticidade real do ativo, da sua exposição e do seu papel no ambiente operacional.

No entanto, a análise da literatura e do contexto organizacional mostra também uma limitação importante. A *CMDB*, por si só, não resolve o problema da gestão integrada de vulnerabilidades e do planeamento de *assessments*, pois não oferece, de forma nativa, mecanismos para consolidar resultados de vulnerabilidades de múltiplas fontes nem para transformar essa informação em regras executáveis de planeamento.

Isto torna-se mais evidente quando a organização mantém processos paralelos fora da *CMDB*, como folhas *Excel* para a reconciliação de inventário e planeamento. Nestas situações, a *CMDB* deixa de funcionar como mecanismo efetivo de governação operacional e passa a servir apenas como fonte parcial de dados. O problema deixa então de ser apenas o inventário dos ativos e passa a ser a ausência de uma camada capaz de ligar o inventário, evidência técnica e execução operacional num fluxo coerente e rastreável.

Neste sentido, a *CMDB* deve ser entendida como condição necessária, mas não suficiente, para suportar automação de ponta a ponta no domínio estudado. O seu valor depende da capacidade de a integrar com mecanismos de consolidação de vulnerabilidades e com processos de planeamento orientados ao risco. É precisamente nessa lacuna que se enquadra a motivação para os módulos propostos neste trabalho.

2.1.3.1 ITIL

A *ITIL* constitui uma das referências mais utilizadas na gestão de serviços de TI, propondo um conjunto de práticas orientadas à organização, operação e melhoria contínua dos serviços (Axelos, 2020). No contexto deste trabalho, a sua relevância decorre sobretudo do papel que atribui à gestão de configuração, de ativos, de mudanças e à rastreabilidade operacional.

A *ITIL* reforça a necessidade de manter o inventário fiável, processos consistentes e ligações claras entre ativos, alterações, incidentes e atividades operacionais. Estes princípios são particularmente importantes em ambientes regulados, onde a organização precisa de demonstrar controlo sobre os seus ativos e sobre os processos que incidem sobre eles.

Contudo, a *ITIL* define princípios e práticas de governação, mas não define mecanismos técnicos concretos para consolidar vulnerabilidades de múltiplas fontes ou para gerar planeamento automatizado de *assessments*. Esta diferença entre governação recomendada e execução técnica constitui uma das razões que justificam a necessidade de uma solução complementar, orientada à automação.

2.1.3.2 Gestão de Ativos (Assets)

Na *ITIL*, um ativo corresponde a qualquer componente que contribui para a entrega de um serviço de TI. A gestão de ativos assegura que esses elementos permanecem identificados, registados e controlados ao longo do seu ciclo de vida, desde a aquisição até à desativação.

Num contexto corporativo como o da Euronext, a *CMDB* mantém informação sobre aplicações, servidores, bases de dados, instâncias *cloud* e *endpoints*, bem como sobre as relações entre esses elementos, como se verifica na Figura 1. Esta estrutura permite compreender dependências técnicas e perceber de que forma uma vulnerabilidade, uma falha ou uma mudança pode afetar serviços críticos.

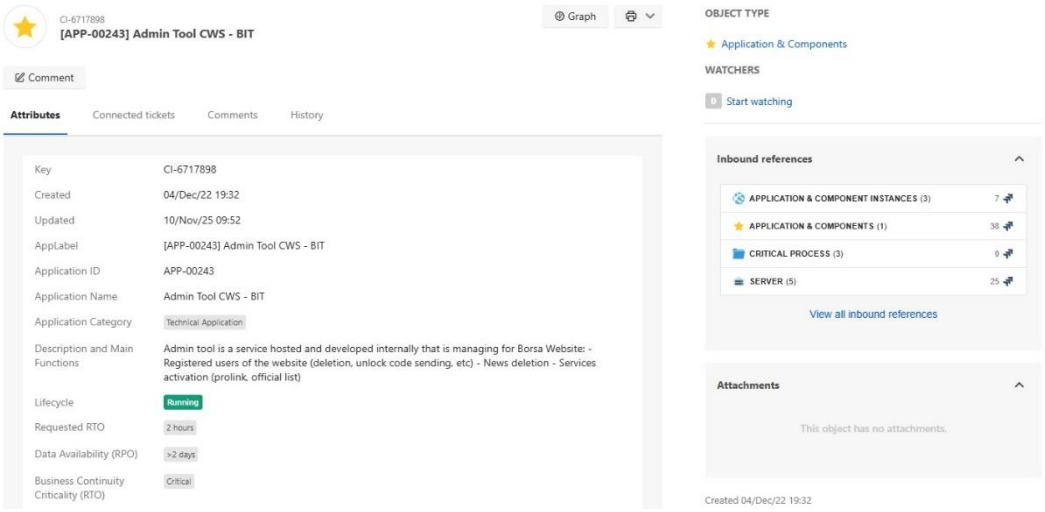


Figura 1 – Exemplo de um *Asset*

A gestão de ativos é relevante para a segurança porque fornece o contexto necessário para priorizar avaliações e interpretar riscos. Um ativo exposto à internet, com elevada criticidade e forte dependência operacional, não deve ser tratado da mesma forma que um ativo interno, com baixo impacto e sem histórico relevante.

No entanto, a gestão de ativos perde valor quando o inventário não reflete relações reais entre componentes ou quando a criticidade existe apenas como atributo documental, sem efeito direto no planeamento. Sem modelação consistente de dependências, exposição e contexto operacional, a organização tende a aplicar esforço excessivo sobre alguns ativos e cobertura insuficiente sobre outros.

2.1.3.3 Gestão de Alterações (Changes)

A gestão de alterações constitui um dos processos centrais da *ITIL* e da *CMDB*, procurando assegurar que modificações à infraestrutura tecnológica são avaliadas, aprovadas e registadas de forma controlada. A sua principal utilidade reside na capacidade de relacionar mudanças com ativos e de antecipar impacto antes da implementação.

No contexto da segurança operacional, esta relação é importante porque alterações em aplicações, servidores ou componentes de infraestrutura podem modificar exposição, introduzir novas vulnerabilidades ou justificar novas avaliações de segurança. A existência de uma CMDB integrada permite, por isso, associar mudanças ao contexto técnico relevante e preservar a rastreabilidade entre alterações e ativos afetados.

2.1.3.4 Testes e Incidentes de Segurança (ISTs)

Os *ISTs (Information Security Tests)* correspondem a atividades de segurança, como *vulnerability assessments* e *penetration tests*, executadas para verificar exposição, controlo e resiliência dos ativos. Em contextos regulados, estes testes assumem importância acrescida porque produzem evidência necessária para demonstrar execução, cobertura e acompanhamento das ações corretivas.

A *CMDB* pode apoiar este processo ao manter ligação entre ativo, teste realizado, resultados obtidos e estado das ações subsequentes. Esta associação melhora a rastreabilidade e reduz a dispersão da informação entre equipas e plataformas.

Contudo, a evidência gerada pelos *ISTs* perde valor quando o planeamento, a execução e os resultados residem em sistemas separados e sem sincronização efetiva. Quando o planeamento vive em folhas *Excel* e a execução vive em *tickets* isolados, a rastreabilidade passa a depender da disciplina manual das equipas e degrada-se com facilidade. Esta limitação reforça a necessidade de mecanismos que mantenham o planeamento e a execução no mesmo fluxo operacional.

2.1.3.5 Gestão de Vulnerabilidades (VULNs)

A gestão de vulnerabilidades corresponde ao processo contínuo de identificação, avaliação, priorização e mitigação de vulnerabilidades em sistemas e aplicações (Souppaya & Scarfone, 2022). As boas práticas publicadas por entidades como a *ENISA* defendem a integração com múltiplas fontes, normalização de dados, classificação de risco e priorização apoiada pelo contexto do ativo (ENISA, 2023).

Esta perspetiva é importante porque a severidade isolada da vulnerabilidade não traduz, por si só, a prioridade operacional da resposta. Uma abordagem útil exige informação adicional sobre a criticidade do ativo, exposição, dependências e impacto no negócio. Sem esse contexto, a organização corre o risco de transformar severidade técnica em prioridade operacional de forma inadequada.

A principal limitação das abordagens tradicionais reside no facto de muitos *scanners* e repositórios tratarem a vulnerabilidade como unidade principal de análise, enquanto o contexto organizacional do ativo permanece incompleto ou inconsistente. Isso dificulta a priorização orientada ao risco e torna mais difícil ligar *findings* técnicos ao planeamento de avaliações de segurança.

2.1.3.6 Análise Crítica – Limitações da CMDB no contexto da segurança operacional

A literatura e as boas práticas reconhecem a *CMDB* como base estrutural para inventário e governação de ativos, mas a sua utilização isolada não resolve os desafios associados à gestão

dinâmica de vulnerabilidades e ao planeamento de testes de segurança. A maioria das implementações concentra-se na modelação de ativos, relações e estados operacionais, mas não integra, de forma nativa, mecanismos de consolidação multi-fonte, deduplicação de vulnerabilidades ou lógica de planeamento orientada ao risco.

Esta limitação é particularmente relevante em organizações onde a evidência técnica se encontra distribuída por múltiplas plataformas e onde o planeamento continua dependente de reconciliação manual fora da *CMDB*. Nesses contextos, a *CMDB* fornece contexto, mas não produz decisão operacional.

Assim, embora constitua uma base essencial, a *CMDB* necessita de uma camada adicional de automação e inteligência que transforme inventário estático em suporte à decisão. É precisamente essa lacuna que motiva o desenvolvimento do *SAS*, orientado à consolidação de vulnerabilidades, e do *RTF*, orientado ao planeamento estruturado de *assessments*.

2.1.4 Processo de deteção de vulnerabilidades

A deteção de vulnerabilidades constitui uma atividade central na segurança operacional, porque permite identificar exposições técnicas relevantes antes da sua exploração e suporta a definição de prioridades de mitigação. Em contextos regulados, esta atividade assume ainda maior importância, porque a organização precisa de demonstrar controlo sobre os seus ativos, evidência sobre o estado de exposição e capacidade de resposta coerente com os requisitos de resiliência digital.

No contexto da Euronext, este processo assenta atualmente numa combinação de inventário mantido na *CMDB*, consultas manuais a várias plataformas de segurança e consolidação posterior dos resultados em folhas *Excel* e *tickets* do *Jira*. O processo inicia-se com a identificação dos ativos a analisar, normalmente a partir do inventário corporativo ou de contexto associado a uma *change* ou a um teste de segurança. Seguidamente, os analistas consultam individualmente várias ferramentas de segurança, recolhem resultados, filtram duplicações, interpretam a severidade e registam a evidência no respetivo fluxo operacional.

Este modelo de funcionamento introduz várias limitações. Em primeiro lugar, distribui a evidência por múltiplas fontes técnicas, exigindo reconciliação manual para produzir uma visão coerente por ativo. Em segundo lugar, torna a qualidade do resultado dependente da experiência do analista e dos critérios utilizados em cada ciclo de análise. Em terceiro lugar, reduz a atualidade da informação, porque os resultados consolidados envelhecem rapidamente quando dependem de exportação, manipulação intermédia e atualização posterior de tickets.

Estas limitações não decorrem de falta de ferramentas especializadas. Pelo contrário, decorrem do facto de cada ferramenta observar apenas uma parte do ecossistema tecnológico e de não existir, no processo atual, uma camada única de consolidação orientada ao ativo e ao contexto da *CMDB*. A seguir analisam-se as principais fontes utilizadas pela Euronext no ciclo de deteção

de vulnerabilidades e as razões pelas quais, apesar da sua utilidade individual, não resolvem isoladamente o problema em estudo.

2.1.4.1 AWS Security Hub

O *AWS Security Hub* funciona como serviço central de agregação de *findings* no ecossistema *AWS*, reunindo alertas provenientes de diferentes serviços da plataforma, como o *Amazon Inspector* (Manne, 2024). No contexto da Euronext, esta ferramenta é utilizada sobretudo para identificar vulnerabilidades e problemas de postura de segurança associados a instâncias EC2 e outros recursos geridos diretamente em *AWS*, como mostra a Figura 2.

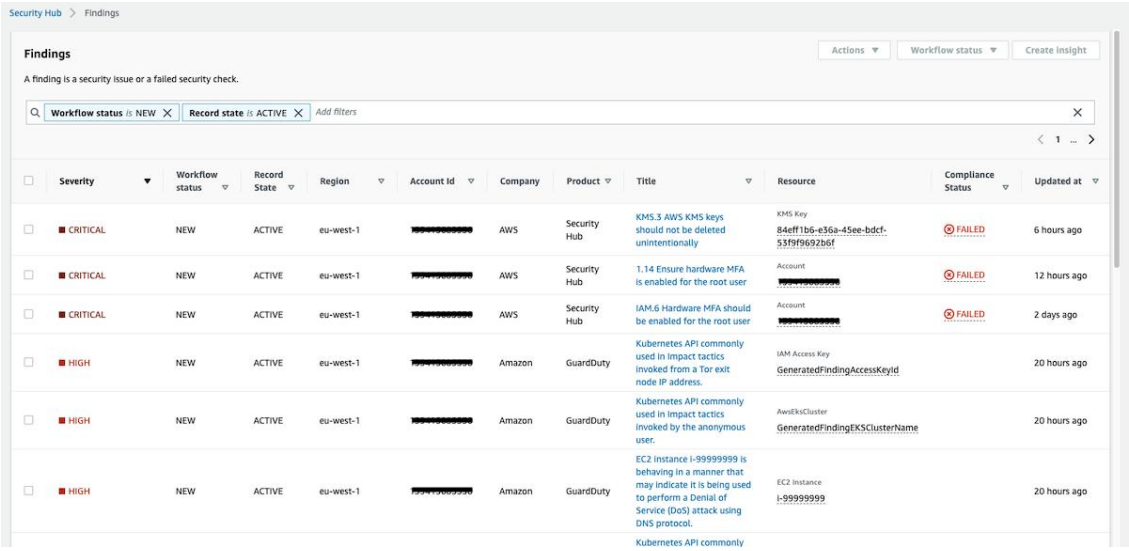


Figura 2 – Vulnerabilidades no *AWS Security Hub*

O valor desta ferramenta reside na capacidade de concentrar informação técnica relevante sobre ativos *cloud* num único ponto de consulta dentro do universo *AWS*. A plataforma fornece *findings* sobre vulnerabilidades, não conformidades e alertas relacionados com o estado de segurança dos recursos observados.

Contudo, esta cobertura permanece limitada ao ecossistema *AWS* e não resolve a necessidade de consolidar informação com outras fontes utilizadas pela organização. Além disso, a interpretação operacional dos resultados continua dependente de associação manual ao inventário da *CMDB* e da sua reconciliação com outras plataformas.

Assim, embora o *AWS Security Hub* forneça evidência útil, não resolve por si só o problema de consolidação multi-fonte orientada ao ativo.

2.1.4.2 Red Hat Insights

O *Red Hat Insights* é uma plataforma de análise proativa orientada a sistemas *Red Hat Enterprise Linux*, permitindo identificar vulnerabilidades, problemas de configuração e necessidades de remediação associadas a ambientes *RHEL* (Chatterjee, 2021). Na Euronext, esta ferramenta é particularmente relevante para a monitorização de servidores *Linux* e para a correlação entre sistemas afetados e vulnerabilidades publicadas sob a forma de *Common Vulnerabilities and Exposures* (*CVEs*), como podemos ver na Figura 3.

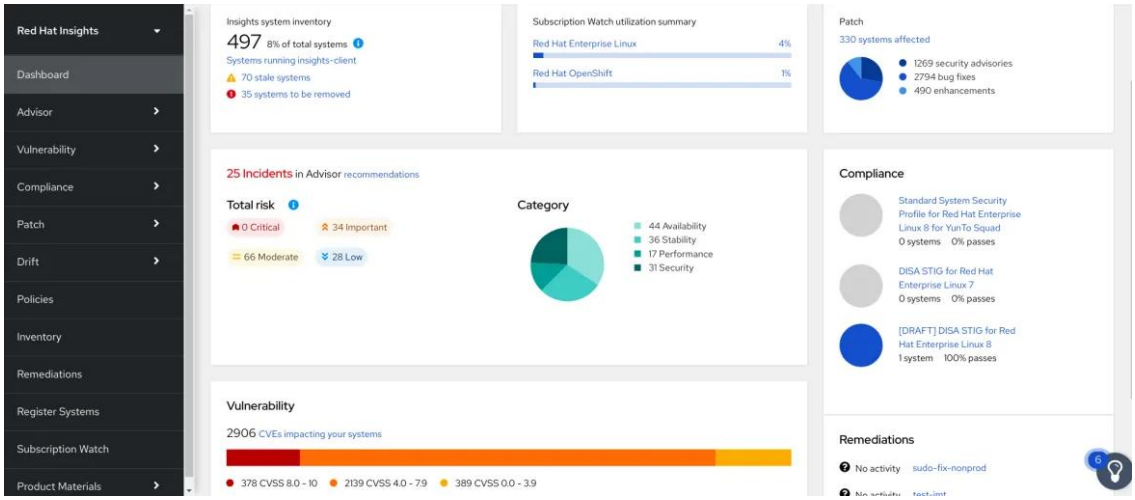


Figura 3 – *Red Hat Insights Dashboard*

A utilidade desta plataforma reside na especialização sobre um subconjunto técnico muito concreto do inventário. Esta especialização melhora a profundidade da análise e fornece recomendações úteis para remediação, sobretudo em ambientes *Linux* empresariais.

No entanto, essa mesma especialização constitui também a sua limitação no contexto global do problema. O *Red Hat Insights* não fornece uma visão transversal sobre o inventário organizacional nem integra, de forma nativa, a informação necessária para o planeamento de *assessments* em articulação com a criticidade do ativo, o seu contexto operacional ou a existência de outras fontes de evidência. O seu contributo é relevante, mas parcial.

2.1.4.3 Microsoft Defender for Cloud

O *Microsoft Defender for Cloud* combina capacidades de *Cloud Security Posture Management* e *Cloud Workload Protection Platform*, permitindo identificar vulnerabilidades, configurações incorretas e riscos operacionais em ambientes *Azure*, híbridos e *multi-cloud* (Diogenes & Janetscheck, 2022). Na Euronext, esta ferramenta é utilizada para observar exposição em máquinas virtuais, bases de dados, redes virtuais e outros recursos associados ao ecossistema *cloud* gerido pela *Microsoft* (Figura 4).

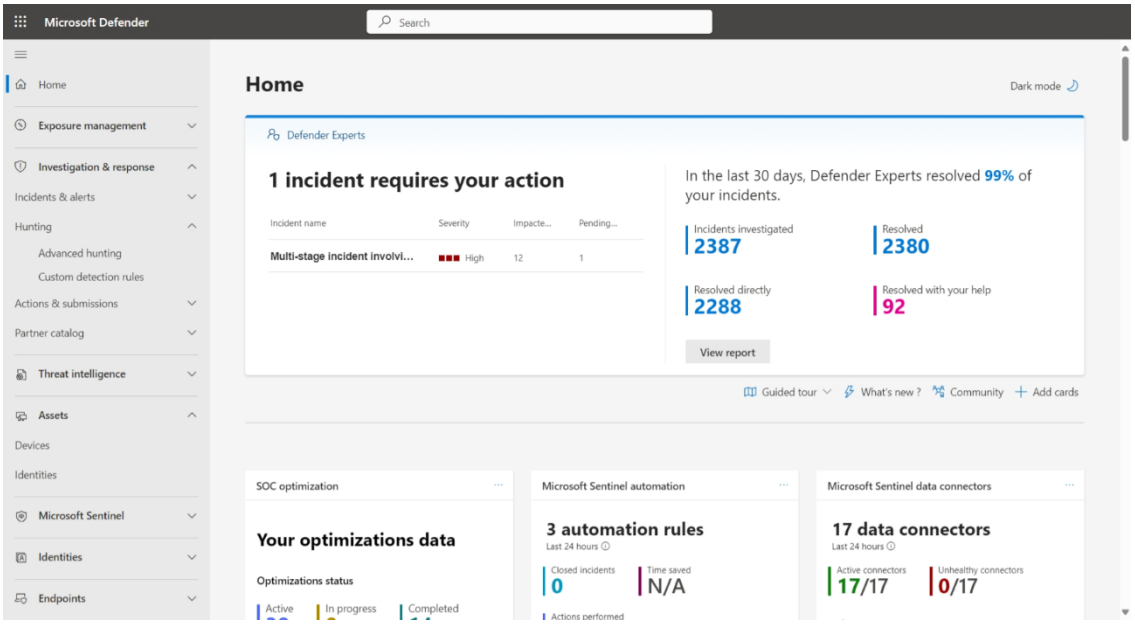


Figura 4 – *Microsoft Defender for Cloud Dashboard*

A sua principal vantagem reside na capacidade de relacionar vulnerabilidades com postura de segurança e configuração, oferecendo uma perspetiva mais ampla do que a simples deteção de *software flaws*. Este tipo de informação é relevante para compreender risco técnico em ambientes distribuídos e híbridos.

Apesar disso, o *Microsoft Defender for Cloud* continua limitado ao âmbito técnico da plataforma e não resolve, isoladamente, a necessidade de consolidar resultados com outras fontes, eliminar duplicações ou transformar *findings* técnicos em decisões de planeamento. A existência de recomendações e vulnerabilidades numa ferramenta não elimina a necessidade de uma camada adicional de correlação com inventário, criticidade e ciclo operacional.

2.1.4.4 Microsoft Defender for Endpoint

O *Microsoft Defender for Endpoint* concentra-se na deteção de ameaças, incidentes, comportamento anómalo e vulnerabilidades específicas de *endpoints Windows*, *Linux* e *macOS* (Huijbregts et al., 2023). Na Euronext, esta ferramenta é utilizada para monitorizar servidores, postos de trabalho e *software* instalado nos *endpoints*, acrescentando visibilidade sobre um conjunto de ativos que não é coberto pelas restantes plataformas da mesma forma (Figura 5).

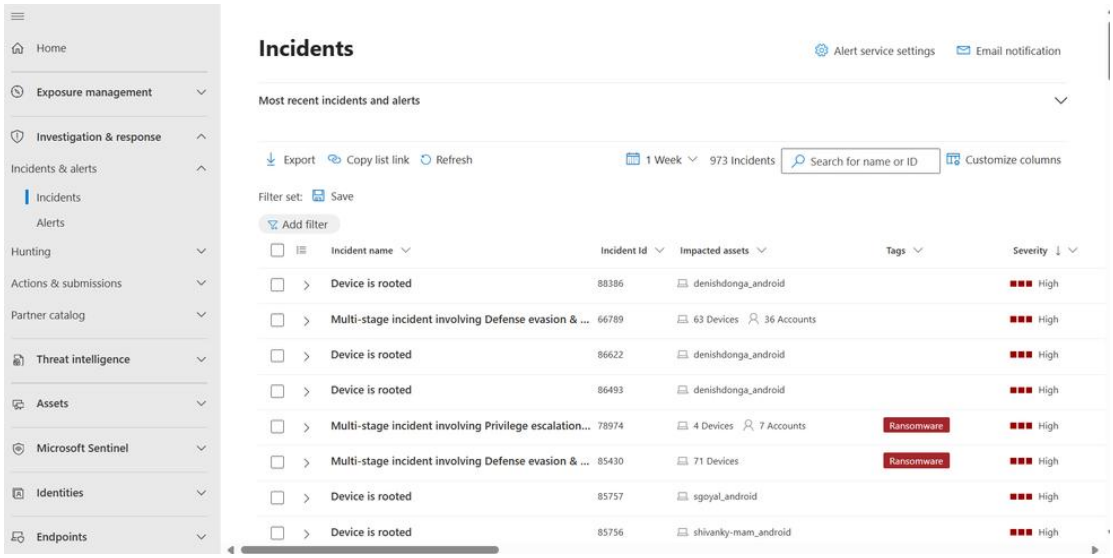


Figura 5 – *Microsoft Defender for Endpoint Incidents*

Esta solução é importante porque introduz uma dimensão diferente da observação de segurança: não apenas em estado de configuração e vulnerabilidade, mas também comportamento em tempo real e contexto associado ao *endpoint*. Em ambientes empresariais, esta visibilidade é particularmente útil para identificar risco operacional em sistemas com grande superfície de exposição.

No entanto, tal como as restantes ferramentas analisadas, o seu valor permanece localizado. A plataforma oferece profundidade sobre *endpoints*, mas não produz uma visão integrada do inventário organizacional nem gera, por si só, um planeamento estruturado de avaliações de segurança. A sua informação precisa de ser reconciliada com outras fontes e com a *CMDB* para adquirir valor operacional mais amplo.

2.1.4.5 Security Scorecard

O *Security Scorecard* adota uma perspetiva externa da postura de segurança, baseada em sinais observáveis da infraestrutura pública e em métricas de exposição acessível a partir da internet (Sohval, 2020). Na Euronext, esta ferramenta é utilizada para avaliar a exposição externa, reputação de *IPs*, *hygiene DNS* e indicadores agregados de postura de segurança (Figura 6).

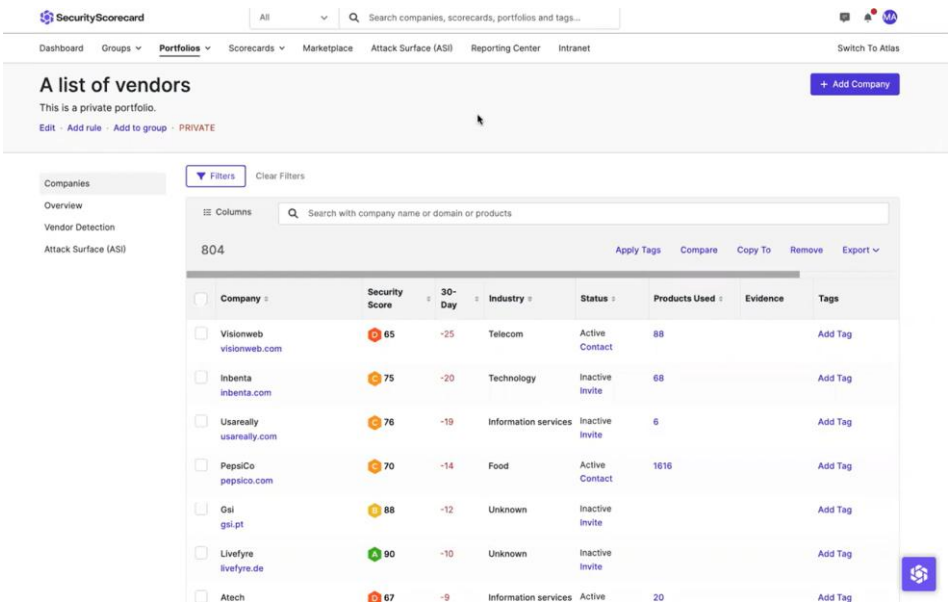


Figura 6 – *Security Scorecard Portfólio*

A utilidade desta abordagem está na capacidade de fornecer uma perspetiva complementar às ferramentas internas. Enquanto outras plataformas observam ativos e configurações a partir do interior do ecossistema organizacional, o *Security Scorecard* observa a organização a partir da sua presença externa. Esta diferença é relevante para identificar riscos de exposição pública e para apoiar decisões sobre ativos mais sensíveis ao contexto *internet-facing*.

Contudo, essa mesma natureza externa limita a granularidade e a associação direta com ativos internos da *CMDB*. A informação produzida é valiosa como sinal adicional de risco, mas não substitui mecanismos de correlação com inventário, criticidade interna ou ciclo de *assessments*. O *Scorecard* complementa a análise, mas não resolve o problema de consolidação e planeamento.

2.1.4.6 NVD

A *National Vulnerability Database* (*NVD*), mantida pelo *National Institute of Standards and Technology* (*NIST*), constitui uma das principais fontes públicas de vulnerabilidades conhecidas, respetivas *CVEs* e métricas de severidade associadas, incluindo *CVSS* (Souppaya & Scarfone, 2022). No contexto da Euronext, a *NVD* é utilizada como fonte complementar para validação de vulnerabilidades identificadas noutras ferramentas e para obtenção de informação técnica adicional sobre *CVEs* específicas.

O valor da *NVD* reside na sua função de referência comum. A sua utilização permite validar classificações, enriquecer informação recolhida noutras plataformas e apoiar um processo mais consistente de interpretação de severidade.

No entanto, a *NVD* não é uma plataforma de gestão de vulnerabilidades, nem uma solução orientada ao contexto organizacional do ativo. A base fornece informação pública sobre vulnerabilidades, mas não sabe onde essas vulnerabilidades se manifestam no inventário da organização, nem qual o seu impacto operacional concreto. Por isso, a *NVD* funciona como mecanismo de enriquecimento e validação, mas não como solução central para o problema em estudo.

2.1.4.7 Análise Crítica – Fragmentação e ausência de integração sistémica

A análise do processo atual e das ferramentas utilizadas mostra que a Euronext dispõe de fontes técnicas relevantes para a identificação de vulnerabilidades, cobrindo diferentes segmentos do seu ecossistema tecnológico. Existe, portanto, maturidade ao nível da recolha de evidência técnica. O problema central não reside na inexistência de ferramentas, mas na fragmentação entre elas e na ausência de uma camada única de consolidação orientada ao ativo.

Cada plataforma analisada observa apenas uma parte do ecossistema, utiliza estruturas próprias de representação e produz resultados que exigem interpretação adicional antes de se tornarem úteis para o planeamento operacional. A coexistência de ambientes *cloud*, *on-premises* e *third-party* agrava esta fragmentação, porque obriga a equipa de segurança a consultar ferramentas distintas e a reconciliar manualmente os resultados.

A literatura evidencia o progresso importante na identificação e classificação de vulnerabilidades, mas tende a assumir cenários em que a consolidação já existe ou em que uma ferramenta central resolve a maior parte do problema. Esse pressuposto não se verifica no contexto estudado. O caso analisado exige uma solução capaz de recolher informação de múltiplas fontes, normalizar resultados heterogéneos, eliminar duplicações e relacionar a evidência técnica com o inventário corporativo mantido na *CMDB*.

É precisamente nesta lacuna que se enquadra o papel do *SAS*. O seu valor não decorre da substituição das ferramentas existentes, mas da capacidade de funcionar como camada de abstração, consolidação e enriquecimento orientada ao inventário corporativo. A necessidade do *SAS* resulta, assim, menos da insuficiência individual das ferramentas e mais da ausência de integração sistémica entre elas.

2.1.5 Processo de planeamento de assessments

O planeamento de *assessments* de segurança constitui uma atividade essencial para garantir a resiliência operacional da Euronext e assegurar o cumprimento de requisitos regulatórios e normativos, nomeadamente os definidos pelo DORA e pelas práticas de gestão de serviços ITIL.

Atualmente, este processo é totalmente manual e depende de várias folhas *Excel* que funcionam como suporte operacional para identificar os ativos a testar, definir periodicidade, distribuir trabalho entre a equipa de InfoSec e comunicar o plano às outras equipas internas.

O planeamento em *Excel* permite flexibilidade imediata, mas cria dependência de regras implícitas e validação manual, o que compromete repetibilidade e auditabilidade. Além disso, o planeamento perde qualidade quando a informação muda na *CMDB* e o *Excel* não reflete essa mudança no mesmo ciclo.

2.1.5.1 Extração e Preparação de Ativos

O processo inicia-se com a extração manual dos ativos diretamente da *CMDB* onde se encontra registado o inventário operacional da Euronext. Esta extração é feita mensalmente, devido às constantes alterações nos *assets* e ao aparecimento de novos.

O responsável pelo planeamento verifica manualmente se os ativos exportados da *CMDB* coincidem com o inventário interno da equipa de segurança, representado numa folha *Excel*, assinalando divergências e classificando criticidade, ciclo de vida e exposição. Esta folha inclui uma tabela com variáveis/colunas já existentes na *CMDB* e outras manuais, tais como:

- CI Key (ex.: CI-6719042);
- Check CI – *Flag* que indica se o *asset* é novo ou não;
- Application Name;
- Lifecycle – Running, Retired, etc.;
- Criticality – Low, Medium, High, Critical;
- Requested RTO;
- Third party application;
- Internet facing;
- And others.

Esta tabela, demonstrada na Figura 7, funciona como o inventário inicial para todo o processo de planeamento.

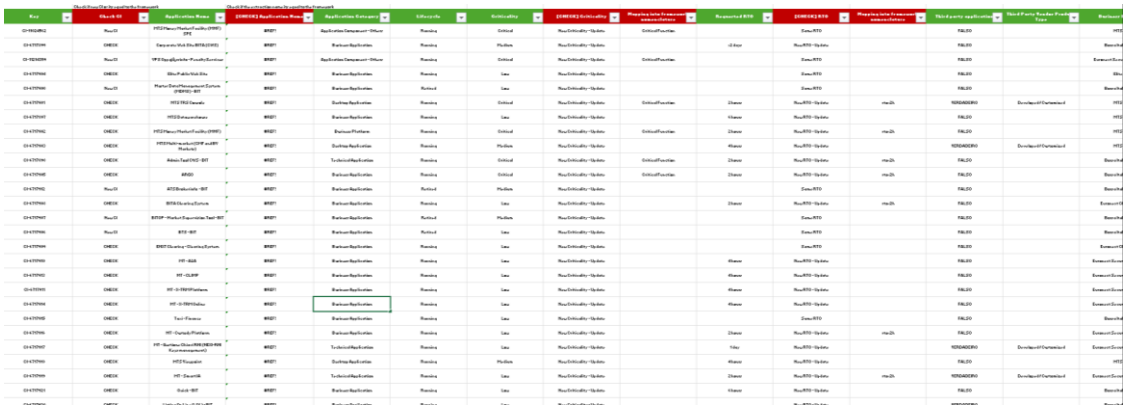


Figura 7 – Extração de Ativos

A extração mensal cria um ponto fixo de desatualização, porque ativos mudam entre ciclos e o planeamento mantém dados antigos. Este desfasamento reduz confiança no inventário operacional e aumenta a probabilidade de omissões ou duplicações de *assessments*.

2.1.5.2 Construção Manual do Plano de Assessments

As folhas seguintes do *Excel*, (Figura 8 e Figura 9) constituem o núcleo do planeamento. O responsável pelo planeamento utiliza estas folhas para atribuir periodicidade, selecionar janelas temporais e distribuir tarefas por analistas e equipas externas.

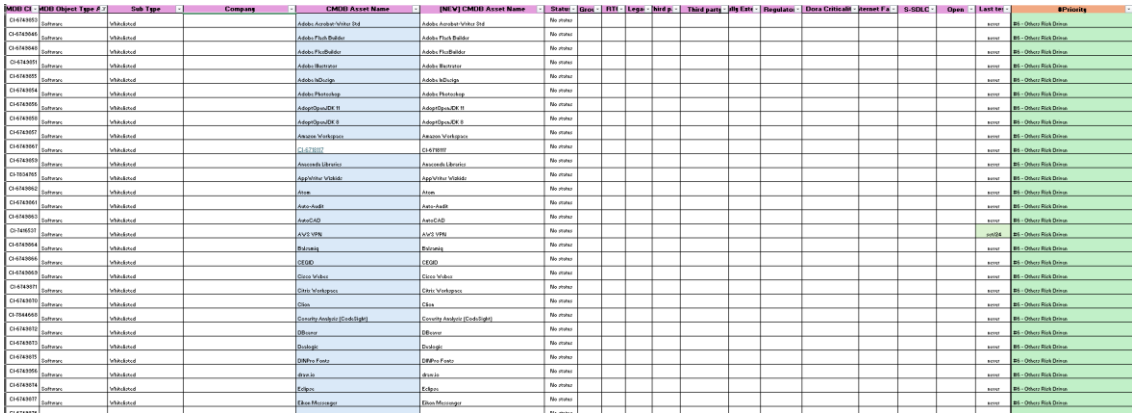


Figura 8 – Plano de *Assessments* (Parte 1)

Entre as colunas mais relevantes encontram-se:

- **Periodicity** – varia entre 12 e 36 meses;
- **Type of Test** – Vários tipos de testes de segurança (*Vulnerability Assessments*, *Penetration Tests*, *Source Code Review*, etc.);
- **Assignee** – Pessoa responsável pela realização do *assessment*;

- **Schedule (# Assessments)** – Anos entre 2025 e 2028 (planeamento para 4 anos), divididos por quadrimestres (ou *quarters*). A data do *assessment* é retratada no respetivo *quarter* através de um '**X**';

- **Infosec Effort / Stakeholders Effort** – Representa o esforço em dias úteis previsto para a realização dos testes e validação por parte dos stakeholders desse ativo.

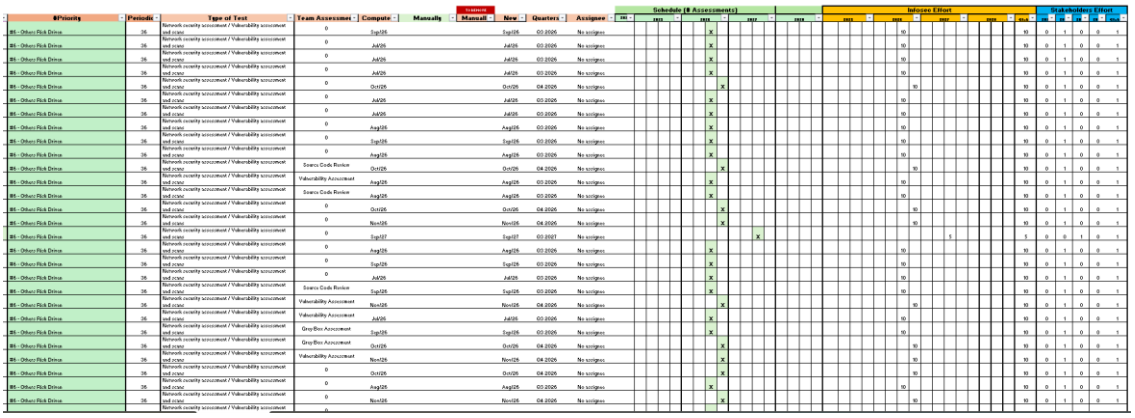


Figura 9 – Plano de *Assessments* (Parte 2)

Após o preenchimento da grelha, os analistas criam tickets no *Jira* para cada *assessment*, com o seu ativo associado e conforme o seu tipo de teste e data de realização.

O uso de "X" por *quarter* simplifica a visualização, mas não preserva racional de decisão nem permite replaneamento automatizado. Sem regras explícitas e executáveis, o planeamento torna-se difícil de reproduzir e difícil de justificar em auditoria.

2.1.5.3 Dashboard Operacional e Controlo de Capacidade

O *Excel* inclui ainda folhas dedicadas à visualização do planeamento e à estimativa de capacidade da equipa. As tabelas e gráficos apresentados mostram:

- Total de testes planeados por ano (2025–2028);

- Distribuição por tipo de teste;

- Distribuição trimestral e mensal;

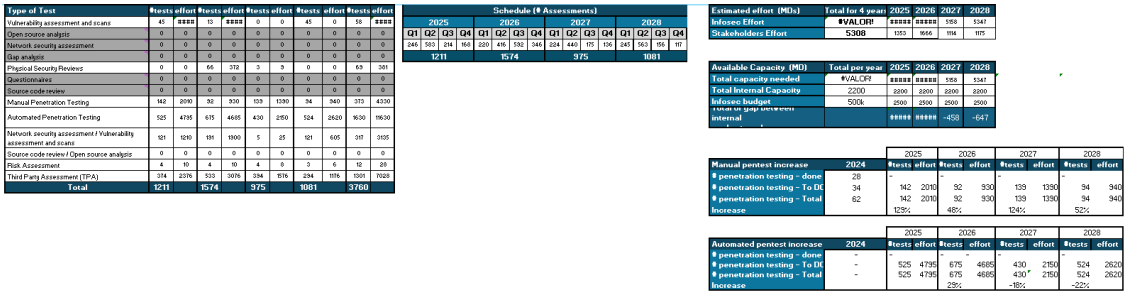
- Carga atribuída a cada analista ou equipa externa;

- Capacidade total da equipa *vs* necessidade de esforço.

Exemplos observados na Figura 10:

- 1211 testes previstos para 2025;

- 1574 para 2026;
- 975 para 2027;
- 1081 para 2028;



Type of Test	#tests	effort	#tests	effort	#tests	effort	#tests	effort	#tests	effort
Vulnerability assessment and scans	45	####	15	####	0	0	45	0	58	####
Open source analysis	0	0	0	0	0	0	0	0	0	0
Network security assessment	0	0	0	0	0	0	0	0	0	0
Gap analysis	0	0	0	0	0	0	0	0	0	0
Physical Security Reviews	0	0	66	372	3	5	0	0	69	381
Questionnaires	0	0	0	0	0	0	0	0	0	0
Source code review	0	0	0	0	0	0	0	0	0	0
Manual Penetration Testing	142	2010	32	330	103	1390	34	940	373	4330
Automated Penetration Testing	525	4795	675	4685	430	2150	524	2620	1630	14630
Network security assessment / Vulnerability assessment and scans	121	1210	131	1300	5	25	121	605	317	3105
Source code review / Open source analysis	0	0	0	0	0	0	0	0	0	0
Risk Assessment	4	10	4	10	4	8	3	6	12	29
Third Party Assessment (TPA)	374	2376	533	3076	334	1576	294	1176	1301	7028
Total	1211		1574		975		1081		3760	

Schedule (# Assessments)															
2025				2026				2027				2028			
Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
246	593	214	158	220	416	530	346	224	440	175	136	243	563	155	117
1211				1574				975				1081			

Estimated effort (MDs)	Total for 4 years	2025	2026	2027	2028
Infosec Effort	#VALOR!	#####	#####	5156	5347
Stakeholders Effort	5308	1353	1666	1114	1175

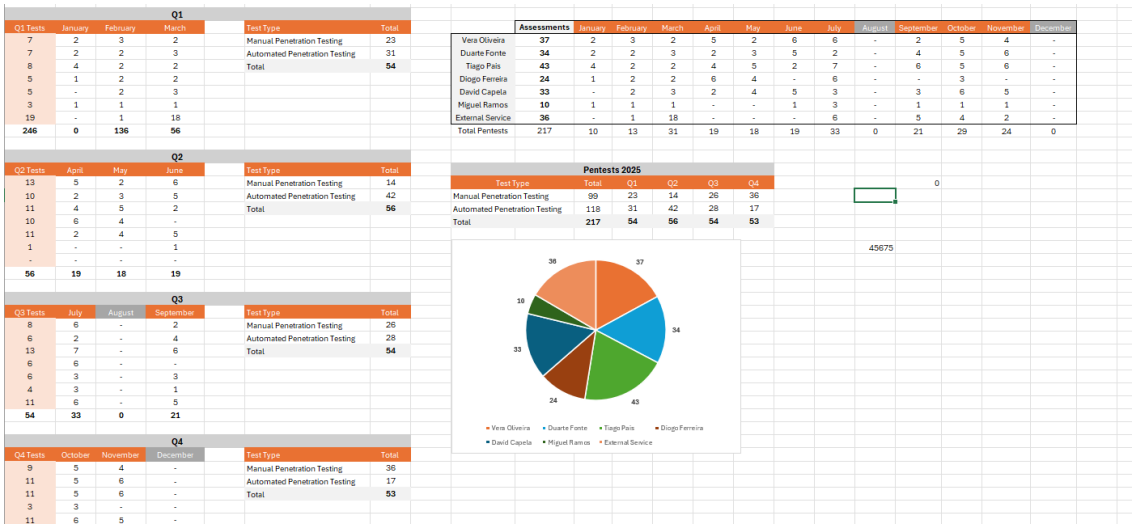
Available Capacity (MD)	Total per year	2025	2026	2027	2028
Total capacity needed	#VALOR!	#####	#####	5156	5347
Total Internal Capacity	2200	2200	2200	2200	2200
Infosec budget	500k	2500	2500	2500	2500
Internal		#####	#####	-459	-647

Manual pentest increase	2024	2025		2026		2027		2028	
		#tests	effort	#tests	effort	#tests	effort	#tests	effort
# penetration testing - done	28	-		-		-		-	
# penetration testing - To DO	34	142	2010	92	930	139	1390	94	940
# penetration testing - Total	62	142	2010	92	930	139	1390	94	940
Increase		129%		48%		124%		52%	

Automated pentest increase	2024	2025		2026		2027		2028	
		#tests	effort	#tests	effort	#tests	effort	#tests	effort
# penetration testing - done	-	-		-		-		-	
# penetration testing - To DO	-	525	4795	675	4685	430	2150	524	2620
# penetration testing - Total	-	525	4795	675	4685	430	2150	524	2620
Increase				29%		-16%		-22%	

Figura 10 – Dashboard com dados analíticos (Parte 1)

Os gráficos circulares mostram ainda a distribuição de *pentests* por analista, entre outras informações, como podemos ver na Figura 11.



Q1					
Q1 Tests	January	Febuary	March	Test Type	Total
7	2	3	2	Manual Penetration Testing	23
7	2	2	3	Automated Penetration Testing	31
8	4	2	2	Total	54
5	1	2	2		
5	-	2	3		
3	1	1	1		
19	-	1	18		
246	0	136	56		

Q2					
Q2 Tests	April	May	June	Test Type	Total
13	5	2	6	Manual Penetration Testing	14
10	2	3	5	Automated Penetration Testing	42
11	4	5	2	Total	56
10	6	4	-		
11	2	4	5		
1	-	-	1		
-	-	-	-		
56	19	18	19		

Q3					
Q3 Tests	July	August	September	Test Type	Total
8	6	-	2	Manual Penetration Testing	26
6	2	-	4	Automated Penetration Testing	28
13	7	-	6	Total	54
6	6	-	-		
6	3	-	3		
4	3	-	1		
11	6	-	5		
54	33	0	21		

Q4					
Q4 Tests	October	November	December	Test Type	Total
9	5	4	-	Manual Penetration Testing	36
11	5	6	-	Automated Penetration Testing	17
11	5	6	-	Total	53
3	3	-	-		
11	6	5	-		

Assessments	January	February	March	April	May	June	July	August	September	October	November	December
Vera Oliveira	37	2	3	2	5	2	6	6	-	2	5	4	-
Duarte Fonte	34	2	2	3	2	3	5	2	-	4	5	6	-
Tiago Pais	43	4	2	2	4	5	2	7	-	6	5	6	-
Diogo Ferreira	24	1	2	2	6	4	-	6	-	-	3	-	-
David Capela	33	-	2	3	2	4	5	3	-	3	6	5	-
Miguel Ramos	10	1	1	1	-	-	1	3	-	1	1	1	-
External Service	36	-	1	18	-	-	-	6	-	5	4	2	-
Total Pentests	217	10	13	31	19	18	19	33	0	21	29	24	0

Pentests 2025					
Test Type	Total	Q1	Q2	Q3	Q4
Manual Penetration Testing	99	23	14	26	36
Automated Penetration Testing	118	31	42	28	17
Total	217	54	56	54	53

Figura 11 – Dashboard com dados analíticos (Parte 2)

Tudo isto é calculado com fórmulas locais do *Excel*, muitas vezes quebradas, sem ligação aos dados reais da *CMDB* nem aos tickets existentes no *Jira*.

Dashboards baseados em fórmulas locais não garantem consistência nem versionamento, e erros silenciosos propagam decisões erradas. Sem ligação a dados reais do *Jira* e *CMDB*, a gestão de capacidade mede intenção e não mede execução.

2.1.5.4 Roadmap por Empresa (Gantt Operacional)

Este *roadmap* é representado noutro ficheiro *Excel*, sem qualquer ligação com o primeiro.

Esta ferramenta é usada para comunicar o planeamento às equipas técnicas, organizar testes paralelos por linha de negócio, controlar conflitos entre janelas de testes e demonstrar o *roadmap* à gestão.

A sua manutenção é também manual e qualquer alteração no planeamento principal não atualiza automaticamente este ficheiro.

Este *Excel*, conforme a Figura 12, fornece uma visualização tipo *Gantt*, organizada por:

- Empresa
- Semanas e meses
- Blocos coloridos representando janelas de execução
- Responsáveis identificados entre parênteses

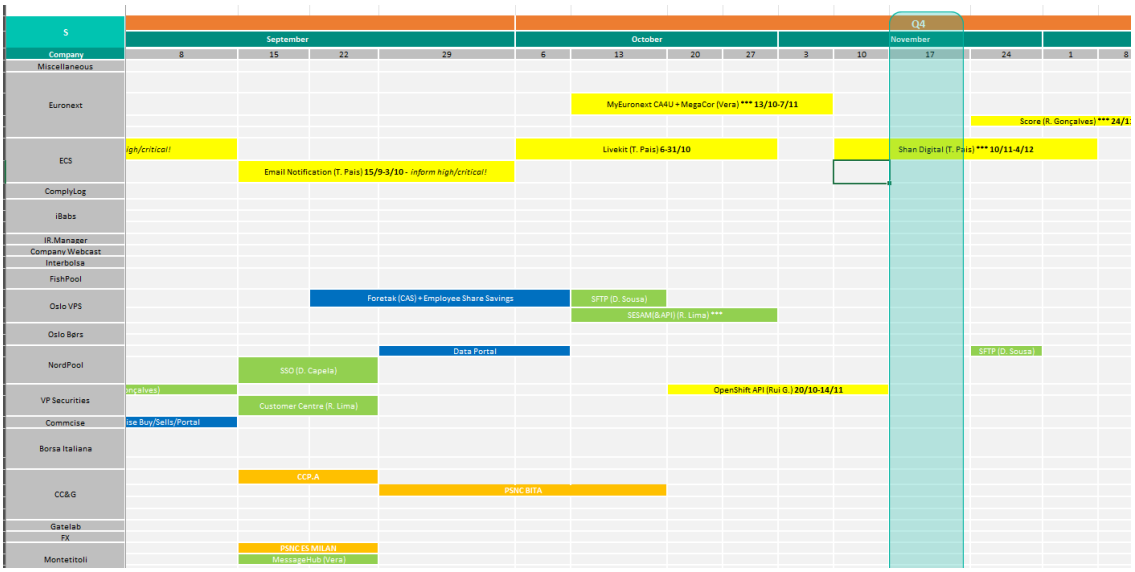


Figura 12 – *Roadmap* de *assessments*

A existência de dois ficheiros sem sincronização cria versões concorrentes de verdade, o que aumenta o risco de conflito entre equipas e janelas de teste. Esta duplicação reduz agilidade, porque cada alteração exige atualização manual em múltiplos pontos.

2.1.5.5 Análise Crítica – Planeamento baseado em folhas Excel e ausência de lógica orientada ao risco

Embora existam recomendações normativas que enfatizam a necessidade de planeamento periódico de avaliações de segurança, a literatura apresenta menos contributos concretos sobre a automação desse planeamento em ambientes empresariais complexos. Na prática industrial, observa-se frequentemente a utilização de folhas *Excel* como mecanismo

operacional de calendarização, o que introduz fragilidade, dependência do fator humano e dificuldade de rastreabilidade.

Modelos tradicionais de planeamento tendem a basear-se em periodicidades fixas, como anual ou bienal, sem incorporar automaticamente variáveis como a criticidade dinâmica do ativo, exposição à internet, histórico de vulnerabilidades abertas ou data do último *assessment* realizado.

Esta ausência de planeamento orientado ao risco representa uma lacuna estrutural que o *RTF* procura mitigar, introduzindo uma lógica automatizada de cálculo de periodicidade e geração de *assessments* baseada em critérios objetivos e atualizados de forma contínua.

2.1.6 Plataformas de Security Orchestration, Automation and Response (SOAR)

As plataformas de *Security Orchestration, Automation and Response* (*SOAR*) emergiram como resposta à crescente complexidade dos ecossistemas de segurança e à necessidade de reduzir o tempo médio de resposta a incidentes (Kovacevic & DiCola, 2023). Estas plataformas integram múltiplas ferramentas de segurança, centralizam alertas provenientes de múltiplas fontes e executam *playbooks* automatizados com o objetivo de padronizar procedimentos operacionais e diminuir a intervenção manual.

A literatura descreve as soluções *SOAR* como extensões naturais de arquiteturas de *Security Information and Events Management* (*SIEM*), permitindo automatizar tarefas de enriquecimento de eventos, correlação de dados e execução de ações de contenção (Rajapakse et al., 2022). No entanto, a sua aplicação é tradicionalmente orientada para a gestão operacional de incidentes em tempo real, e não necessariamente para o planeamento estruturado de avaliações de segurança ou integração profunda com *CMDBs* corporativas.

Com o objetivo de analisar a adequação destas plataformas ao problema identificado neste trabalho, foram consideradas duas soluções amplamente reconhecidas no mercado: ***Palo Alto Cortex XSOAR*** e ***Splunk SOAR***.

2.1.6.1 Palo Alto Cortex XSOAR

O *Cortex XSOAR*, desenvolvido pela *Palo Alto Networks*, é uma plataforma de orquestração e automação de segurança que integra centenas de ferramentas através de conectores nativos. Permite a criação de *playbooks* automatizados que executam tarefas como enriquecimento de alertas, correlação de eventos, bloqueio automático de *IPs* maliciosos ou recolha de evidências.

A arquitetura do *Cortex XSOAR* assenta numa abordagem centralizada, onde eventos provenientes de diferentes fontes são normalizados e tratados segundo *workflows* configuráveis. A plataforma inclui funcionalidades de gestão de casos, colaboração entre equipas e *dashboards* operacionais.

Contudo, apesar da sua capacidade de integração ampla e automatização de tarefas recorrentes, o *Cortex XSOAR* está primariamente orientado para a resposta a incidentes e gestão de alertas, não oferecendo, de forma nativa, mecanismos estruturados de planeamento de *vulnerability assessments* baseados em criticidade de ativos ou histórico de testes. A integração com *CMDBs* pode ser realizada através de *APIs* mas não constitui o foco central da plataforma nem está intrinsecamente ligada a modelos de planeamento orientado ao risco.

2.1.6.2 Splunk SOAR

O *Splunk SOAR* segue uma abordagem semelhante, permitindo a orquestração de ações automatizadas através de *playbooks* visuais. A plataforma é frequentemente utilizada em conjunto com o *Splunk SIEM*, facilitando a automação da resposta a incidentes identificados através de análise de *logs* e eventos.

O *Splunk SOAR* permite criar fluxos automatizados que executam tarefas como consulta a serviços externos, enriquecimento de dados de ameaças, abertura automática de *tickets* e execução de ações de contenção. A sua principal vantagem reside na capacidade de reduzir o tempo médio de resposta (*MTTR*) e de padronizar procedimentos operacionais.

Todavia, tal como no caso do *Cortex SOAR*, a sua lógica é predominantemente reativa, centrada na gestão de eventos e incidentes. A plataforma não disponibiliza um modelo orientado à consolidação sistemática de múltiplas fontes de vulnerabilidades com deduplicação avançada, nem um mecanismo de cálculo automatizado de periodicidade de testes com base na criticidade dos ativos registados numa *CMDB* corporativa.

2.1.6.3 Análise Crítica – Plataformas SOAR no Contexto do Projeto

A análise das plataformas *SOAR* evidencia que estas soluções são particularmente eficazes na automação de resposta a incidentes e na orquestração de ações operacionais em tempo real. No entanto, o problema identificado no contexto da *Euronext* apresenta características diferentes.

O desafio central não reside apenas na resposta a eventos de segurança, mas na consolidação estruturada de vulnerabilidades provenientes de múltiplas fontes heterogéneas, na sua normalização e deduplicação, e na articulação dessas informações com o inventário corporativo mantido na *CMDB*.

O projeto requer um mecanismo de planeamento sistemático de *assessments* de segurança, baseado em critérios como criticidade do ativo, exposição, histórico de vulnerabilidades e data do último teste. Esta dimensão de planeamento estratégico e calendarização orientada ao risco não constitui o foco primário das plataformas *SOAR* analisadas.

Por fim, a adoção de uma plataforma *SOAR* implicaria frequentemente adaptação dos processos internos aos modelos da ferramenta, potenciais custos elevados de licenciamento e dependência tecnológica de um fornecedor específico. Num contexto onde o objetivo é preservar ferramentas existentes e introduzir uma camada modular de integração e automação, uma solução especializada como o *Security Command Hub* revela-se mais alinhada com as necessidades identificadas.

Embora partilhem princípios de automação e integração, as plataformas *SOAR* não substituem a necessidade de uma arquitetura modular composta por um serviço dedicado à consolidação e enriquecimento de vulnerabilidades (*SAS*) e por uma *framework* orientada ao planeamento estruturado de testes (*RTF*). A proposta do *Security Command Hub* posiciona-se, assim, como uma solução complementar e mais ajustada ao problema específico analisado, integrando inventário corporativo, gestão de vulnerabilidades e planeamento orientado ao risco num ecossistema coerente.

2.2 Resultados

Após a pesquisa efetuada, o presente subcapítulo apresenta os resultados da pesquisa, detalhando as lacunas e limitações do processo atual, conclusões extraídas sobre integração com *CMDBs* e descrevendo os dois módulos propostos: o Security Automation Service (SAS) e o Resilience Testing Framework (RTF).

Para cada módulo, são analisadas as integrações, a arquitetura e os mecanismos de orquestração.

2.2.1 Lacunas e limitações do processo atual

A pesquisa acima evidencia que a maturidade de uma plataforma de gestão de vulnerabilidades depende de três capacidades essenciais:

- Inventário fiável de ativos;
- Recolha contínua e consolidada de evidências;
- Mecanismos consistentes de priorização e planeamento de mitigação.

Em ambientes regulados, estas capacidades assumem uma grande importância por suportarem rastreabilidade, auditabilidade e controlo operacional, aspetos reforçados por requisitos como os definidos pelo *DORA* (European Union, 2022; European Commission, 2024).

No contexto analisado, o processo atual apresenta lacunas que afetam diretamente a eficiência operacional e a qualidade de decisão da equipa de segurança da Euronext.

Em primeiro lugar, a gestão do inventário, que apesar de estar suportada por uma *CMDB*, depende de extrações manuais e de reconciliação com folhas *Excel*, o que cria a possibilidade de divergências e reduz a fiabilidade do inventário operacional ao longo do tempo. A literatura de *ITSM* e ITIL reforça que a *CMDB* deve funcionar como fonte central e atualizada de informação para suportar processos críticos, evitando a fragmentação de dados e a duplicação de registos (Axelos, 2020; Agutter, 2020).

Em segundo lugar, a deteção e consolidação de vulnerabilidades decorrem de forma descentralizada, com consultas independentes a múltiplas plataformas. Este modelo aumenta o esforço humano necessário, dificulta a consistência entre analistas e favorece redundâncias e desatualização rápida dos dados, especialmente quando as fontes não partilham modelos de dados compatíveis. Esta limitação contrasta com as recomendações de boas práticas que apontam para a necessidade de processos contínuos e repetíveis, suportados por automatização e *governance* de dados (ENISA, 2023; Souppaya & Scarfone, 2022).

Em terceiro lugar, a priorização tende a ficar condicionada pela variabilidade das fontes e por critérios pouco uniformes. Embora métricas como o *CVSS* sejam amplamente utilizadas para representar severidade, a literatura mostra que a priorização eficaz requer contextualização e critérios adicionais, como por exemplo, a exposição do ativo, a criticidade operacional e informação complementar de risco, com o risco de produzir listas extensas de baixo valor operacional (Walkowski et al., 2021; Souppaya & Scarfone, 2022). Em paralelo, ferramentas de avaliação externa, como o *Security ScoreCard*, oferecem indicadores úteis de postura de segurança, mas não resolvem o problema de correlação com ativos internos e execução de planos de mitigação quando usadas de forma isolada (Sohval, 2020).

Por fim, a pesquisa revela que a ausência de integração entre a gestão de vulnerabilidades e os ciclos de operação compromete a escalabilidade e capacidade de resposta. Abordagens modernas de *DevOps* e *DevSecOps* defendem automatização e integração de forma a reduzir falhas humanas e a aumentar a previsibilidade operacional em contextos complexos (Bass et al., 2015; Rajapakse et al., 2022; Kim et al., 2021). Esta necessidade torna-se ainda mais crítica quando o objetivo é produzir evidências rastreáveis de controlo de risco e de testes periódicos.

2.2.2 Integração com a CMDB

Os resultados da pesquisa evidenciam que a integração com a *CMDB* é um fator crítico para a automação e *governance* dos processos de segurança, particularmente em ambientes regulados e de elevada complexidade operacional. Mais uma vez, a literatura e as boas práticas de *ITSM* indicam que a *CMDB* deve funcionar como fonte única de verdade, suportando não apenas inventário, mas também processos de segurança, auditoria e planeamento (Axelos, 2020; Agutter, 2020).

No contexto do *Jira Service Management*, a *CMDB* é implementada através do módulo *Assets*, que disponibiliza um modelo de dados extensível para representar *Configuration Items* (*CIs*) e respetivas relações. A pesquisa técnica realizada demonstra que esta plataforma suporta a

integração programática completa através de *REST APIs*, permitindo operações de leitura, criação, atualização e eliminação de objetos, bem como a associação direta desses objetos a *issues* do *Jira* (Atlassian, 2024).

Um aspeto central identificado é a necessidade de garantir a consistência bidirecional dos dados entre o sistema de segurança e a *CMDB*. Por um lado, a informação existente na *CMDB* – como criticidade, estado do ativo, relações e responsáveis – deve ser consumida pelos módulos de segurança para suportar decisões de priorização e planeamento. Por outro lado, os resultados das avaliações de segurança, como vulnerabilidades identificadas, datas de testes ou estados de mitigação, devem ser refletidos automaticamente no *Jira*, através de associação com *issues* de testes ou de vulnerabilidade, evitando discrepâncias entre sistemas e assegurando sempre a ligação da informação e rastreabilidade contínua.

A pesquisa evidencia que a *Assets Query Language* (*AQL*) desempenha um papel fundamental neste processo. A *AQL* permite a execução de consultas avançadas sobre a *CMDB*, possibilitando a seleção dinâmica de ativos com base em atributos, relações e estados. Alguns exemplos incluem a extração de todos os ativos críticos expostos à internet (*internet facing*), filtragem de ativos associados a determinada empresa, entre muitas outras. O uso de *AQL* elimina a necessidade de exportações manuais e permite que os processos de segurança operem diretamente sobre o inventário da organização (Atlassian, 2024; Alén, 2020).

Outro resultado relevante da pesquisa relaciona-se com a disponibilidade de bibliotecas e *SDKs* que facilitam a integração da *API* do *Jira* em diferentes linguagens de programação, podendo estas influenciar o desenvolvimento e manutenção a longo prazo pela Euronext. A documentação e o ecossistema da plataforma indicam o suporte direto ou indireto para linguagens amplamente utilizadas em ambientes empresariais, como por exemplo *Java* e *Python*, através de bibliotecas oficiais ou criadas pela comunidade. Esta diversidade tecnológica permite que a integração seja realizada de forma consistente com arquiteturas modernas, reduzindo o esforço de desenvolvimento e riscos de manutenção (Atlassian, 2024).

Adicionalmente, a integração com a *CMDB* e o sistema de *issues* do *Jira* permite associar ativos a *changes*, *tasks* de testes e de vulnerabilidades, garantindo rastreabilidade bidirecional entre inventário e execução operacional. A pesquisa destaca que esta capacidade é essencial para suportar auditorias e provas de conformidade, aspetos referidos no *DORA*.

2.2.3 Security Automation Service (SAS)

Os resultados da pesquisa mostram que a automação da gestão de vulnerabilidades exige um componente dedicado à recolha, consolidação, normalização e enriquecimento de informação proveniente de múltiplas fontes. Esta necessidade não decorre da ausência de ferramentas de segurança, mas da ausência de um mecanismo intermédio capaz de transformar evidência técnica dispersa numa estrutura coerente e utilizável no contexto do inventário corporativo e do planeamento operacional.

Foi neste enquadramento que surgiu o *SAS*, concebido como camada de integração entre as fontes de deteção de vulnerabilidades e os sistemas internos da organização, em particular a *CMDB* e o *RTF*. O papel do *SAS* não consiste em substituir as ferramentas existentes, mas em abstrair as suas diferenças, consolidar os resultados produzidos e devolver informação preparada para consumo operacional.

A sua função principal consiste em recolher dados brutos de diferentes plataformas, aplicar processos de normalização, eliminar duplicações e enriquecer os registos com métricas de risco e metadados relevantes. Esta abordagem responde diretamente às lacunas identificadas no processo atual, nomeadamente fragmentação de evidência, heterogeneidade dos formatos e dificuldade em relacionar *findings* técnicos com o contexto do ativo registado na *CMDB*.

Ao mesmo tempo, o *SAS* permite operar de forma autónoma como motor de pesquisa orientado ao ativo, permitindo consulta direta sobre um elemento específico do inventário. Esta possibilidade é relevante porque reforça o valor do serviço não apenas como componente de integração com o *RTF*, mas também como mecanismo independente de apoio à análise técnica.

2.2.3.1 Integrações

A análise das fontes utilizadas no contexto da Euronext mostra que a eficácia de um serviço de automação de vulnerabilidades depende diretamente da sua capacidade de integração programática com múltiplas plataformas de segurança. As ferramentas consideradas neste trabalho disponibilizam *APIs* com diferentes níveis de maturidade, permitindo acesso a vulnerabilidades, recomendações, severidades e metadados associados aos ativos observados.

No contexto do *SAS*, estas integrações não representam apenas conectividade técnica. Representam a condição necessária para construir uma visão consolidada sobre o estado de exposição dos ativos. O problema estudado não se resolve com acesso isolado a uma única *API*, mas com a capacidade de articular várias fontes que observam segmentos diferentes do ecossistema tecnológico da organização.

As integrações prioritárias consideradas para o *SAS* incluem *AWS Security Hub*, *Red Hat Insights*, *Microsoft Defender for Cloud*, *Microsoft Defender for Endpoint*, *Security Scorecard* e *NVD*. Em conjunto, estas fontes cobrem diferentes domínios do inventário, incluindo ambientes *cloud*, sistemas *Linux*, *endpoints*, exposição externa e bases de dados públicas de vulnerabilidades.

O *AWS Security Hub* disponibiliza *APIs* orientadas à consulta de *findings*, estados de conformidade e resultados agregados sobre recursos *AWS*, permitindo filtrar informação por severidade, recurso afetado e estado de resolução (Manne, 2024). Esta capacidade torna-o relevante para a recolha estruturada de vulnerabilidades associadas ao universo *AWS*, mas o valor operacional dos seus resultados depende da sua correlação com o inventário corporativo e com outras fontes.

O *Red Hat Insights* fornece *APIs* que expõem vulnerabilidades associadas a sistemas *Red Hat Enterprise Linux*, bem como recomendações de remediação e informação ligada a *CVEs* (Chatterjee, 2021). Estas *APIs* são particularmente úteis para ambientes *Linux* empresariais, mas a sua especialização sobre um subconjunto técnico do inventário reforça a necessidade de consolidação com outras fontes.

No ecossistema *Microsoft*, a pesquisa distingue duas superfícies de integração com funções diferentes. O *Microsoft Defender for Cloud* disponibiliza *APIs* orientadas à postura de segurança em ambientes *cloud* e híbridos, incluindo recomendações, controlos e vulnerabilidades associadas a *workloads* (Diogenes & Janetscheck, 2022). O *Microsoft Defender for Endpoint* disponibiliza *APIs* específicas para *endpoints*, focadas em ameaças, incidentes e vulnerabilidades associadas a *software* instalado (Huijbregts et al., 2023). A separação entre estas duas superfícies reflete a diversidade funcional das fontes e evidencia a necessidade de uma camada de normalização intermédia.

O *Security Scorecard* disponibiliza *APIs* orientadas à obtenção de métricas externas de postura de segurança, pontuações agregadas e indicadores de exposição pública (Sohval, 2020). Esta informação acrescenta uma perspetiva complementar às restantes fontes, mas a sua utilidade operacional depende da capacidade de a relacionar com ativos internos e com critérios de criticidade mantidos na *CMDB*.

Por fim, a *NVD* disponibiliza *APIs* públicas para consulta de *CVEs*, métricas *CVSS* e referências técnicas associadas (Souppaya & Scarfone, 2022). O seu papel no contexto do *SAS* não consiste em substituir as restantes fontes, mas em complementar e enriquecer os dados recolhidos, fornecendo uma referência padronizada para validação de severidade e contextualização técnica das vulnerabilidades.

A integração com estas *APIs* introduz desafios transversais, como heterogeneidade dos modelos de dados, diferenças na granularidade da informação, mecanismos distintos de autenticação, limitações de quota e variabilidade na qualidade dos resultados devolvidos. Estes desafios não são acessórios. São precisamente a razão pela qual a solução proposta exige uma camada intermédia dedicada.

2.2.4 Resilience Testing Framework (RTF)

O planeamento sistemático de avaliações de segurança constitui um dos principais desafios operacionais em organizações com grande volume de ativos e com exigências regulatórias elevadas, como a Euronext. A literatura e as boas práticas de gestão de serviços e segurança mostram que a eficácia de *vulnerability assessments*, *penetration tests* e outras avaliações não depende apenas da execução técnica de cada teste, mas também da existência de processos consistentes de planeamento, priorização, calendarização e acompanhamento ao longo do tempo (Axelos, 2020; European Union, 2022).

Neste enquadramento, o *RTF* surge como o componente responsável por estruturar o ciclo de planeamento e gestão de *assessments*. O seu papel central consiste em transformar o inventário mantido na *CMDB* em decisões operacionais sobre quando, como e com que prioridade cada ativo deve ser avaliado. Para esse efeito, o *RTF* utiliza atributos como criticidade, exposição, tipo de ativo, contexto regulamentar e histórico de testes, aproximando planeamento técnico e risco operacional.

A relevância desta abordagem é reforçada pelo *DORA*, que exige a realização periódica de testes de resiliência digital e a manutenção de evidência clara sobre o seu planeamento, execução e resultados (European Union, 2022; European Commission, 2024). O problema, contudo, não reside apenas na obrigação de testar. Reside na ausência de mecanismos que transformem essa obrigação em regras executáveis, consistentes e auditáveis dentro do ecossistema operacional da organização.

A pesquisa mostra também que uma *framework* de planeamento eficaz não deve limitar-se a um único tipo de avaliação. Deve suportar múltiplos tipos de *assessments*, incluindo *vulnerability assessments*, *penetration tests*, *source code reviews* e outras avaliações especializadas, cada uma com prioridades, periodicidades e condições de execução distintas. Esta diversidade torna inviável um modelo puramente manual quando a organização opera sobre milhares de ativos e múltiplas dependências técnicas.

Outro resultado relevante da pesquisa relaciona-se com a gestão de capacidade e dependências. A literatura mostra que o planeamento de testes deve considerar não apenas a criticidade do ativo, mas também disponibilidade de recursos, relações entre sistemas e restrições operacionais, de forma a evitar conflitos, redundâncias e sobrecarga nas equipas (Axelos, 2020; Kim et al., 2021). Esta exigência reforça a necessidade de um componente que não apenas registe *assessments*, mas que mantenha contexto suficiente para apoiar decisões repetíveis e coerentes.

A pesquisa reforça ainda a importância da rastreabilidade entre inventário, planeamento, execução e ações corretivas. A associação entre ativos, avaliações planeadas, resultados obtidos e vulnerabilidades identificadas é necessária para suportar auditorias, relatórios de conformidade e melhoria contínua dos processos de segurança (ISO/IEC 27001:2022, 2022). É precisamente nesta articulação entre planeamento e evidência que o RTF adquire valor.

2.2.4.1 Orquestrador de processos

Os resultados da pesquisa mostram que a automação do planeamento de avaliações de segurança exige um mecanismo central de orquestração, capaz de executar processos recorrentes, aplicar regras de negócio de forma consistente e sincronizar estados entre inventário, persistência interna e sistema de trabalho. Esta necessidade está alinhada com abordagens modernas de automação operacional e com práticas *DevOps* e *DevSecOps*, que

favorecem processos recorrentes, executáveis e com baixa dependência de intervenção manual (Bass et al., 2015; Kim et al., 2021; Rajapakse et al., 2022).

No contexto do *RTF*, o orquestrador existe para resolver uma limitação estrutural do processo manual: a ausência de um mecanismo único que mantenha inventário, planeamento, execução e atualização de estados dentro de um fluxo operacional coerente. Em vez de depender de extrações periódicas, fórmulas locais e reconciliação manual, o *RTF* concentra estes processos numa lógica contínua e executável.

O primeiro grupo de processos diz respeito à sincronização e consistência dos dados operacionais. Esta função mantém o inventário persistido alinhado com a *CMDB*, atualiza informação relevante para o planeamento e assegura que a informação proveniente da *CMDB* prevalece sobre cópias locais, preservando o seu papel como referência principal do inventário. Esta decisão está alinhada com os princípios de gestão de configuração descritos na *ITIL*, que reforçam a importância de uma fonte de verdade coerente para suportar processos críticos (Axelos, 2020; Agutter, 2020; Atlassian, 2024).

É também nesta fase que são calculados atributos operacionais necessários ao planeamento, como prioridade, tipo de teste e a data sugerida de avaliação. Estes cálculos traduzem regras explícitas sobre criticidade, exposição, vulnerabilidades abertas e histórico de *assessments*, permitindo que o planeamento deixe de depender apenas de interpretação humana e passe a assentar em critérios executáveis.

Um segundo grupo de processos diz respeito à reavaliação de vulnerabilidades já identificadas. A pesquisa em gestão de vulnerabilidades e *patch management* destaca a importância de validar continuamente se uma vulnerabilidade permanece efetivamente aberta após as ações corretivas, reduzindo discrepâncias entre registo formal e exposição real (Souppaya & Scarfone, 2022). Neste contexto, o orquestrador do *RTF* suporta fluxos de reteste que articulam *tickets* existentes, ativos associados e verificação técnica através do *SAS*.

Um terceiro grupo de processos suporta a execução automatizada de determinados *assessments*, nomeadamente *Vulnerability Assessments*. Aqui, o papel do orquestrador consiste em transformar planeamento em ação operacional, solicitando ao *SAS* a recolha de evidência técnica e convertendo o resultado obtido em atualização do estado do *assessment* e eventual criação de novas tarefas de vulnerabilidade. Esta lógica aproxima o planeamento de uma execução efetiva e rastreável, respondendo a uma limitação clara do processo manual, onde planeamento e execução vivem em artefactos separados.

Por fim, o orquestrador suporta também mecanismos de comunicação operacional e visibilidade de gestão, como a geração periódica de informação sobre *assessments* planeados. Este tipo de funcionalidade não é apenas conveniência operacional. É também um mecanismo de previsibilidade, coordenação e produção de evidência sobre o próprio processo de planeamento, aspetos particularmente relevantes em ambientes regulados.

Assim, o valor do orquestrador não reside apenas na execução automática de tarefas. O seu contributo central está em transformar um processo manual, disperso e pouco repetível num conjunto de processos coerentes, regulares e alinhados com requisitos de rastreabilidade e conformidade.

2.2.4.2 Integração com o SAS

A integração entre o *RTF* e o *SAS* constitui um elemento central da abordagem proposta, porque liga diretamente planeamento e evidência técnica. Sem esta integração, o *RTF* seria apenas uma ferramenta de calendarização e registo. Com ela, passa a dispor de um mecanismo para acionar recolha de vulnerabilidades e incorporar resultados técnicos no ciclo operacional dos *assessments*.

Esta integração é realizada através de uma *API* dedicada exposta pelo *SAS*, permitindo uma comunicação desacoplada e reutilizável entre os dois componentes. Esta opção é importante do ponto de vista arquitetural, porque preserva independência entre os módulos e permite que cada um evolua segundo as suas responsabilidades específicas. O *SAS* mantém o foco em consolidação e normalização de vulnerabilidades. O *RTF* mantém o foco em planeamento, ciclo de vida dos *assessments* e rastreabilidade operacional.

Do ponto de vista funcional, esta integração permite ao *RTF* solicitar verificações de segurança com base em ativos concretos previamente identificados no contexto da *CMDB*. Cada invocação ao *SAS* parte, assim, de um contexto operacional explícito e não de uma pesquisa genérica desligada do inventário corporativo.

Esta relação entre os dois módulos traduz uma das ideias centrais do trabalho: a separação entre recolha de evidência técnica e decisão de planeamento, mantendo ao mesmo tempo uma articulação clara entre ambas. Em vez de fundir tudo numa única aplicação monolítica, a solução distribui responsabilidades e utiliza integração por *API* para construir um fluxo mais flexível, extensível e alinhado com o ecossistema existente da Euronext.

2.3 Análise

A análise do estado da arte mostra que a gestão de vulnerabilidades e o planeamento de *assessments* continuam a constituir desafios estruturais em organizações de grande dimensão e em contextos regulados, como o da Euronext. A literatura e a prática industrial revelam disponibilidade de ferramentas maduras para inventário, deteção de vulnerabilidades, orquestração operacional e gestão de *tickets*. No entanto, essas capacidades surgem distribuídas por plataformas distintas e raramente convergem num mecanismo único de integração entre inventário, evidência técnica e planeamento orientado ao risco.

Do ponto de vista científico e técnico, a principal lacuna identificada não reside na ausência de métodos para detetar vulnerabilidades nem na ausência de modelos para gerir inventário. A lacuna reside na articulação sistemática entre três dimensões que a literatura tende a tratar separadamente: inventário corporativo mantido numa *CMDB*, evidência técnica recolhida a partir de múltiplas fontes heterogéneas de vulnerabilidades e mecanismos automatizados de planeamento de *assessments*. Esta separação torna-se especialmente problemática em ambientes regulados, onde a organização precisa de transformar informação dispersa em decisões auditáveis, rastreáveis e repetíveis.

Do ponto de vista operacional, a primeira alternativa consiste na manutenção do processo atual, baseado em extrações manuais da *CMDB*, consultas independentes a várias ferramentas de segurança e consolidação posterior em folhas *Excel*. Esta abordagem apresenta a vantagem de exigir baixo esforço inicial de implementação, mas a literatura e a observação do processo atual mostram que não escala adequadamente, introduz inconsistências frequentes e enfraquece a rastreabilidade exigida por contextos de auditoria e por requisitos como os definidos pelo *DORA*.

Uma segunda alternativa consiste na adoção de uma ferramenta única de gestão de vulnerabilidades, capaz de substituir as plataformas atualmente utilizadas. Embora esta opção pareça atrativa do ponto de vista da centralização, revela-se pouco adequada num contexto como o da Euronext. O ecossistema tecnológico em análise combina ambientes *cloud*, *on-premises* e *third-party*, e cada ferramenta existente cobre um subconjunto específico desse ecossistema. A substituição integral por uma única plataforma introduziria custos elevados, perda de especialização funcional e forte dependência de um fornecedor, sem garantir cobertura equivalente sobre todas as áreas do inventário.

Neste contexto, a decisão de preservar as ferramentas existentes e integrá-las através de uma camada dedicada de automação surge como alternativa mais consistente com a realidade organizacional e com as boas práticas identificadas na literatura. Esta abordagem permite manter o valor das plataformas já adotadas, reduzir o impacto de mudança e concentrar o esforço técnico no verdadeiro problema: a ausência de integração sistémica entre inventário, vulnerabilidades e planeamento.

Ao nível arquitetural, a pesquisa identificou também duas alternativas de desenho relevantes. A primeira corresponde à adoção de plataformas comerciais de *SOAR* como solução transversal. Embora estas plataformas ofereçam capacidades importantes de orquestração e automação, o seu foco recai predominantemente sobre incidentes, alertas e resposta operacional. Além disso, a sua adoção implicaria custos elevados, dependência tecnológica adicional e limitações na adaptação a modelos de dados e processos específicos da organização. A segunda alternativa corresponde à implementação de um sistema monolítico, agregando deteção, planeamento e gestão de ativos numa única aplicação. A literatura mostra, no entanto, que arquiteturas monolíticas tendem a dificultar a evolução independente, a escalabilidade seletiva e a integração com fontes externas em domínios sujeitos a mudança contínua.

A Tabela 2 sintetiza a posição do *Security Command Hub* face a duas classes de soluções existentes: plataformas *SOAR* e plataformas de *vulnerability management*. O objetivo desta comparação não consiste em desvalorizar essas soluções, mas em evidenciar que o problema abordado neste trabalho não é totalmente coberto por nenhuma delas quando considerado no contexto específico de integração com *CMDB*, consolidação multi-fonte e planeamento estruturado de *assessments*.

Tabela 2 – Comparação com outras soluções

Critério	Plataformas *SOAR*	Plataformas de Vulnerability Management	Security Command Hub
Foco principal	Orquestração e resposta a incidentes	Identificação e gestão de vulnerabilidades	Integração entre inventário, vulnerabilidade e planeamento de *assessments*
Integração com a *CMDB*	Possível, mas não central	Variável e nem sempre profunda	Central e estrutural
Consolidação multi-fonte orientada ao ativo	Parcial	Parcial	Sim
Normalização e deduplicação contextual	Limitada ao modelo da ferramenta	Dependente da plataforma	Sim
Planeamento de *assessments*	Não é foco principal	Normalmente ausente ou limitado	Sim
Cálculo de prioridade orientado ao risco do ativo	Parcial	Parcial	Sim
Determinação automática do tipo de teste	Não	Não	Sim
Sugestão de agendamento de *assessments*	Não	Não	Sim
Integração com o *Jira* para ciclo operacional	Possível	Variável	Nativa no contexto do projeto
Preservação das ferramentas já existentes	Parcial	Nem sempre	Sim

A comparação evidencia que as soluções analisadas resolvem partes relevantes do problema, mas não oferecem, de forma integrada, a combinação de capacidades exigidas no contexto estudado. As plataformas *SOAR* são fortes na orquestração reativa e na automação operacional em torno de incidentes. As plataformas de *vulnerability management* são fortes na identificação, priorização e remediação de vulnerabilidades. No entanto, nenhuma destas abordagens articula de forma nativa o inventário corporativo mantido na *CMDB*, a consolidação contextual de múltiplas fontes e o planeamento estruturado de *assessments* com base em criticidade, exposição e histórico.

Com base nesta análise, a opção por uma arquitetura modular composta pelo *SAS* e pelo *RTF* torna-se tecnicamente justificada. Esta separação permite isolar responsabilidades, manter desacoplamento entre recolha de evidência técnica e planeamento operacional, e facilitar evolução independente de cada componente. Ao mesmo tempo, a integração por *APIs* com a *CMDB*, com o *Jira* e entre os próprios módulos permite preservar a coerência com o ecossistema existente e responder de forma flexível à introdução de novas fontes, novas métricas e novas regras de planeamento.

Deste modo, o *Security Command Hub* surge como uma resposta específica a uma lacuna identificada tanto na literatura como no processo existente: a inexistência de um mecanismo que transforme inventário corporativo, evidência técnica dispersa e requisitos de rastreabilidade em decisões executáveis de planeamento e em evidência operacional coerente.

3 Proposta de Solução e Design

O presente capítulo tem como objetivo apresentar a solução proposta para responder ao problema identificado nos capítulos anteriores. A proposta baseia-se nas lacunas observadas na gestão de vulnerabilidades, na fragmentação das fontes de informação e nas limitações de integração entre plataformas de segurança e *CMDB*.

Inicialmente são definidos os requisitos funcionais e não funcionais que orientam o desenvolvimento da solução, estabelecendo os princípios que suportam a sua arquitetura e funcionamento. Estes requisitos traduzem necessidades identificadas no processo atual de gestão de vulnerabilidades e no planeamento de *assessments*.

Seguidamente, é apresentada a arquitetura do *Security Command Hub*, descrevendo a organização geral da solução e os dois componentes principais que a compõem: o *Security Automation Service* (*SAS*) e o *Resilience Testing Framework* (*RTF*). Para cada componente são analisadas as suas responsabilidades, integrações e papel no fluxo global do sistema.

O capítulo descreve ainda os modelos de dados que suportam o armazenamento e processamento de informação de vulnerabilidades e de inventário proveniente da *CMDB*. São também apresentados os fluxos operacionais do sistema e os algoritmos responsáveis pela priorização de ativos (e suas vulnerabilidades) e pelo agendamento de *assessments*.

Por fim, são discutidas as principais decisões arquiteturais consideradas durante o *design* da solução, bem como os principais *trade-offs* associados às opções adotadas. Esta análise procura justificar as escolhas realizadas e demonstrar o alinhamento da solução com os objetivos definidos para o projeto.

3.1 Recontextualização do Problema

O estado da arte apresentado no capítulo anterior demonstra que organizações como a Euronext, utilizam múltiplas ferramentas para a deteção de vulnerabilidades, gestão de ativos e execução de avaliações de segurança. No entanto, estas ferramentas operam frequentemente de forma isolada, dificultando a consolidação de informação e a automação de processos.

No contexto analisado, a informação sobre vulnerabilidades encontra-se distribuída por diversas plataformas de segurança, enquanto o inventário corporativo reside na *CMDB*. Esta separação cria dependências operacionais que dificultam a correlação entre vulnerabilidades, ativos e histórico de avaliações de segurança.

A ausência de um mecanismo central de consolidação e normalização de vulnerabilidades impede a criação de uma visão consistente do risco associado aos ativos registados na *CMDB*. Como consequência, as equipas de segurança dependem frequentemente de processos manuais para correlacionar informação proveniente de diferentes fontes.

Esta limitação afeta também o planeamento de *assessments* de segurança, que depende da identificação correta de ativos críticos, da análise do histórico de vulnerabilidades e da definição de prioridades de avaliação. Sem integração estruturada entre inventário e evidência técnica, este processo torna-se difícil de escalar e manter atualizado.

Adicionalmente, a inexistência de mecanismos automatizados de priorização e agendamento reduz a capacidade de alinhar avaliações de segurança com o risco efetivo associado a cada ativo. Esta limitação torna mais difícil garantir cobertura consistente de testes e responder de forma sistemática a requisitos de resiliência operacional.

Face a estas limitações, torna-se necessário conceber uma solução que permita consolidar informação de vulnerabilidades proveniente de múltiplas fontes, integrar essa informação com o inventário corporativo e apoiar o planeamento estruturado de *assessments* de segurança. Esta necessidade motiva o desenvolvimento do *Security Command Hub*, apresentado nos capítulos seguintes.

3.2 Requisitos

A definição dos requisitos da solução baseia-se na análise do processo atual de gestão de vulnerabilidades e planeamento de *assessments* de segurança, complementada com a recolha de informação junto de *stakeholders* envolvidos nas diferentes etapas deste processo.

Esta abordagem permite validar as limitações identificadas no capítulo anterior e compreender de forma mais detalhada as necessidades operacionais das equipas responsáveis pela gestão de ativos, execução de avaliações de segurança e supervisão estratégica da função de cibersegurança.

Para este efeito foram realizadas entrevistas com três pessoas da Euronext (duas delas, *stakeholders* definidos previamente, pertencentes a diferentes subequipas dentro de InfoSec; e o *Chief Information Security Officer* (*CISO*)) que representam perspetivas diferentes do processo.

Estas entrevistas tiveram como objetivo identificar e clarificar dificuldades do processo atual, compreender os diferentes impactos e a necessidade de automação e recolher contributos para a definição dos requisitos funcionais e não funcionais da solução proposta.

3.2.1 Entrevistas com os Stakeholders

Com o objetivo de compreender melhor o funcionamento do processo atual e validar as necessidades operacionais associadas à gestão de vulnerabilidades e ao planeamento de *assessments* de segurança, foram realizadas entrevistas com três intervenientes ou interessados neste processo.

Os participantes incluem um responsável pela gestão de ativos e planeamento de *assessments*, um especialista responsável pela execução de avaliações de segurança e um responsável executivo pela função de cibersegurança da Euronext.

Os dois primeiros participantes representam perspetivas operacionais diretamente relacionadas com o processo de gestão de vulnerabilidades. O terceiro participante representa uma perspetiva estratégica associada à *governance* de segurança, gestão de risco e cumprimento de requisitos regulamentares associados ao *DORA*.

Esta combinação de perspetivas permite analisar o processo atual de forma abrangente, considerando necessidades operacionais das equipas técnicas e requisitos de rastreabilidade e conformidade ao nível organizacional.

Tabela 3 – Entrevistados

Nome	Função	Perspetiva
Rui	Management & Roadmap Team	Gestão de ativos e planeamento
Vera	Application & Code Security	Execução de *assessments*
Paulo	CISO	Estratégia, *compliance* e *DORA*

3.2.1.1 Guião da Entrevista

Para garantir consistência entre as entrevistas realizadas, foi definido um conjunto reduzido de questões abertas relacionadas com o processo atual de gestão de vulnerabilidades e planeamento de *assessments*, integração com a *CMDB* e necessidades de automação.

Tabela 4 – Guião da Entrevista

Pergunta	Objetivo
Como é atualmente realizada a correlação entre vulnerabilidades e ativos na *CMDB*?	Compreender a integração atual
Quais são as principais dificuldades do processo atual de gestão de vulnerabilidades?	Identificar limitações
Que fontes de vulnerabilidades são utilizadas?	Identificar fragmentação
Como é realizado o planeamento de *assessments*?	Entender o processo atual
Que tipo de automação considera mais relevante?	Identificar necessidades
Que tipo de rastreabilidade considera necessária para a auditoria e conformidades regulatórias exigidas pelo DORA?	Validar requisitos do *DORA*

3.2.1.2 Respostas

Este subcapítulo apresenta um resumo das respostas recolhidas durante as entrevistas realizadas. As respostas foram transcritas de forma resumida, preservando o conteúdo essencial das observações apresentadas pelos participantes relativamente ao funcionamento do processo atual e às principais dificuldades identificadas.

As respostas vão ser orientadas por pergunta e não por entrevistado, para facilitar a leitura.

Pergunta 1 – "Como é atualmente realizada a correlação entre vulnerabilidades e ativos registados na *CMDB*?"

Rui: "A equipa começa normalmente pelos relatórios gerados pelas ferramentas de segurança e depois tenta perceber a que ativo corresponde cada vulnerabilidade. Em muitos casos é necessário consultar a *CMDB* para confirmar manualmente o sistema ou a aplicação afetada."

Vera: "Normalmente analisamos os relatórios gerados pelas ferramentas e tentamos identificar a aplicação ou sistema afetado com base na informação técnica disponível. Quando essa correspondência não é clara, é necessário validar a informação com o inventário ou contactar as equipas responsáveis."

Paulo: "A Euronext possui várias ferramentas que produzem informação sobre vulnerabilidades, mas a ligação dessa informação aos ativos registados na *CMDB* ainda depende bastante da análise realizada pelas equipas técnicas."

Pergunta 2 – "Quais são as principais dificuldades do processo atual de gestão de vulnerabilidades?"

Rui: "A maior dificuldade está na dispersão da informação entre as diferentes ferramentas de segurança. A equipa precisa de consolidar manualmente os resultados para compreender que vulnerabilidades afetam cada ativo."

Vera: “Muitas vezes encontramos a mesma vulnerabilidade reportada em diferentes ferramentas. Isso obriga-nos a verificar manualmente se se trata da mesma vulnerabilidade ou de problemas distintos.”

Paulo: “Do ponto de vista de gestão de risco, a principal dificuldade é obter rapidamente uma visão consolidada das vulnerabilidades associadas aos ativos mais críticos.”

Pergunta 3 – “Que fontes de vulnerabilidades são utilizadas no processo atual?”

Rui: “A organização usa diferentes ferramentas de segurança que analisam aplicações, infraestruturas e dependências de software. Cada ferramenta produz relatórios próprios que precisam de ser analisados pela equipa.”

Vera: “Usamos ferramentas de análise de código, *scanners* de dependências e *scanners* de infraestrutura, como por exemplo, o *AWS Security Hub* para tudo o que está relacionado com *AWS*, etc. Cada ativo pode ter uma ferramenta especifica, como no caso da *AWS*, mas também utilizamos bibliotecas de vulnerabilidades genéricas, como o *NVD*, para confirmação de resultados.”

Paulo: “Possuímos várias ferramentas especializadas que produzem informação técnica relevante, mas essa informação está distribuída por diferentes plataformas.”

Pergunta 4 – “Como é realizado o planeamento de *assessments* de segurança?”

Rui: “O planeamento baseia-se na criticidade dos ativos e no histórico de avaliações anteriores. Tentamos perceber que sistemas precisam de novos *assessments*, mas essa informação nem sempre está centralizada.”

Vera: “Normalmente recebemos *assessments* com base na criticidade das aplicações e em resultados de *scanners*. Em alguns casos o planeamento depende também da disponibilidade das equipas, principalmente aqueles que são mais demorados.”

Paulo: “O planeamento dos *assessments* deve garantir a cobertura adequada dos ativos mais críticos e assegurar que avaliações são realizadas com a periodicidade adequada, especialmente quando existem requisitos regulamentares associados à resiliência operacional.”

Pergunta 5 – “Que tipo de automação considera mais relevante?”

Rui: “Seria muito útil ter uma plataforma que ajude a planear *assessments*, permitindo acompanhar as avaliações realizadas, planeadas e prioridades associadas aos ativos mais críticos.”

Vera: “A equipa beneficiaria de uma plataforma que apresente uma visão consolidada das vulnerabilidades associadas a cada ativo.”

Paulo: "A automação deve permitir consolidar informação relevante e apoiar decisões sobre quando e em que ativos devemos realizar *assessments*, garantindo o alinhamento entre o risco identificado e a necessidade de novos *assessments*."

Pergunta 6 – "Que tipo de rastreabilidade considera necessária para a auditoria e conformidades regulatórias exigidas pelo *DORA*?"

Rui: "É importante manter o registo claro das vulnerabilidades identificadas e também das avaliações realizadas em cada ativo, incluindo a data das avaliações e resultados obtidos."

Vera: "A equipa precisa de acompanhar o ciclo completo de uma vulnerabilidade e também compreender quando um ativo foi avaliado pela última vez e que resultados foram obtidos."

Paulo: "A organização precisa de garantir rastreabilidade entre ativos, vulnerabilidades identificadas e avaliações realizadas, assegurando evidência clara para as auditorias e o cumprimento dos requisitos regulamentares do *DORA*."

3.2.2 Requisitos Funcionais

Com base na análise do processo atual e nas observações recolhidas durante as entrevistas, foram identificados um conjunto de requisitos funcionais que orientam o desenvolvimento da solução proposta.

Os requisitos definidos refletem necessidades operacionais identificadas pelas equipas responsáveis pela gestão de vulnerabilidades e pela execução de *assessments*, bem como necessidades de *governance* relacionadas com rastreabilidade e com o cumprimento de requisitos regulamentares, principalmente relacionados com o *DORA*.

A solução proposta organiza-se em dois componentes principais: *SAS* e *RTF*. Cada componente responde a necessidades específicas identificadas durante o levantamento de requisitos.

3.2.2.1 Requisitos do Security Automation Service

O *SAS* tem como objetivo consolidar informação de vulnerabilidades provenientes de múltiplas ferramentas e associar essa informação aos ativos registados na *CMDB*.

Tabela 5 – Requisitos Funcionais do *SAS*

ID	Título	Descrição
RF-SAS-01	Pesquisa de vulnerabilidades por ativo	O sistema deve permitir a pesquisa de vulnerabilidades associadas a um ativo específico com base na informação registada na *CMDB*, consultando as ferramentas relevantes para esse ativo.

RF-SAS-02	Consolidação de vulnerabilidades	O sistema deve recolher vulnerabilidades provenientes das ferramentas de segurança utilizadas pela organização e consolidar essa informação numa estrutura comum que permita uma análise centralizada.
RF-SAS-03	Normalização de informação de vulnerabilidades	O sistema deve normalizar a informação recolhida das diferentes ferramentas de segurança, garantindo consistência na representação de vulnerabilidades, ativos afetados e metadados associados.
RF-SAS-04	Deduplicação de vulnerabilidades	O sistema deve identificar vulnerabilidades duplicadas provenientes de diferentes ferramentas e agrupar esses registos, permitindo reduzir redundância na informação apresentada às equipas de segurança.
RF-SAS-05	Associação de vulnerabilidades a ativos	O sistema deve associar vulnerabilidades identificadas aos ativos registados na *CMDB*, permitindo relacionar resultados das ferramentas de segurança com o inventário corporativo.
RF-SAS-06	Consolidação de informação de risco	O sistema deve disponibilizar uma visão consolidada das vulnerabilidades associadas a cada ativo, permitindo às equipas de segurança compreender rapidamente o nível de exposição de cada sistema.

3.2.2.2 Requisitos do Resilience Testing Framework

O *RTF* tem como objetivo apoiar o planeamento e acompanhamento de *assessments* associados aos ativos registados na *CMDB*.

Tabela 6 – Requisitos Funcionais do *RTF*

ID	Título	Descrição
RF-RTF-01	Planeamento de *assessments* de segurança	O sistema deve permitir planear avaliações de segurança para ativos registados na *CMDB*, considerando fatores como criticidade do ativo, histórico de avaliações e informação de vulnerabilidades associadas.
RF-RTF-02	Gestão de calendário de avaliações	O sistema deve manter o registo estruturado das avaliações realizadas e das avaliações planeadas, permitindo acompanhar o estado dos *assessments* de cada ativo.
RF-RTF-03	Priorização de avaliações de segurança	O sistema deve apoiar a priorização de *assessments* com base na criticidade dos ativos e na informação de vulnerabilidades identificadas.
RF-RTF-04	Histórico de avaliações	O sistema deve manter histórico das avaliações realizadas, incluindo a data da avaliação, ativos avaliados e vulnerabilidades encontradas.

RF-RTF-05	Integração com o sistema de gestão de trabalho	O sistema deve permitir criar e atualizar tickets no *Jira* associados a *assessments*.
RF-RTF-06	Execução automatizada de determinados *assessments*	O sistema deve permitir executar automaticamente determinados tipos de avaliações de segurança, nomeadamente "*Vulnerability Assessments*" recorrendo ao *SAS* e registando os resultados obtidos no ticket correspondente.
RF-RTF-07	Rastreabilidade entre vulnerabilidades e avaliações	O sistema deve permitir relacionar vulnerabilidades identificadas com avaliações de segurança realizadas, garantindo rastreabilidade entre evidência técnica e os *assessments*.
RF-RTF-08	Suporte a auditoria e conformidade	O sistema deve manter a informação que permita demonstrar rastreabilidade entre ativos, vulnerabilidades identificadas e avaliações realizadas, suportando necessidades de auditoria e requisitos regulamentares como os impostos pelo *DORA*.

3.2.3 Requisitos Não Funcionais

Os requisitos não funcionais definem características de qualidade que a solução deve cumprir para garantir funcionamento adequado no contexto organizacional em que será utilizada. Estes requisitos consideram aspetos relacionados com segurança, desempenho, escalabilidade, integração com sistemas existentes e capacidade de auditoria.

Tabela 7 – Requisitos Não Funcionais

ID	Título	Descrição
RNF-01	Integração com sistemas existentes	A solução deve integrar-se com os sistemas já utilizados pela Euronext, incluindo a *CMDB*, o *Jira* e as ferramentas de segurança adotadas.
RNF-02	Escalabilidade	A solução deve suportar crescimento no número de ativos registados na *CMDB* e no volume de vulnerabilidades analisadas, garantindo que o sistema continua a responder de forma consistente à medida que a infraestrutura evolui.
RNF-03	Segurança da Informação	A solução deve garantir proteção da informação processada, incluindo dados de vulnerabilidades, ativos e *assessments*, assegurando o controlo de acesso adequado e a proteção contra utilização não autorizada.
RNF-04	Rastreabilidade e auditoria	O sistema deve manter o registo das ações realizadas, incluindo execuções de *assessments*, vulnerabilidades detetadas e ativos, permitindo

		reconstruir um histórico de atividades relacionadas com os ativos.
RNF-05	Fiabilidade	A solução deve garantir consistência na recolha e processamento de informação proveniente das diferentes ferramentas de segurança, evitando perdas de dados e assegurando que os resultados apresentados refletem a informação disponível nas plataformas acedidas.
RNF-06	Manutenibilidade	A arquitetura do sistema deve permitir adaptação e extensão da solução, facilitando a integração futura de novas ferramentas de segurança ou alterações nos processos de planeamento da Euronext.

3.3 Arquitetura Geral do Security Command Hub

A arquitetura geral do *Security Command Hub* organiza a solução em dois serviços principais, SAS e RTF, que colaboram para apoiar o processo de gestão de vulnerabilidades e planeamento de *assessments*.

A solução segue uma arquitetura baseada em serviços independentes, permitindo separar responsabilidades entre os dois componentes principais. Esta separação facilita a evolução futura da solução e permite escalar cada componente de forma independente quando necessário.

A arquitetura física da solução, aproveitando a infraestrutura disponível da Euronext, utiliza o *Elastic Container Service* (*ECS*) da *AWS*, que permite um ambiente baseado em *containers*. O acesso às aplicações ocorre através de um *Application Load Balancer* (*ALB*) que encaminha os pedidos para os containers correspondentes.

3.3.1 Arquitetura do SAS

O *SAS* é um serviço independente que permite pesquisar vulnerabilidades associadas a um ativo específico, ou a um conjunto deles. No entanto, pode também ser invocado pelo *RTF* durante a execução de determinados tipos de *assessments*.

3.3.1.1 Arquitetura Lógica do SAS

O processo inicia quando o serviço recebe um pedido de análise associado a um ativo ou *ticket* existente no *Jira*. O sistema identifica o(s) ativo(s) relevante(s) e consulta os atributos dos mesmos na *CMDB*.

Estes atributos incluem informações como produto, fabricante, domínio, entre outros, e permitem determinar quais as ferramentas de segurança mais adequadas para recolher a informação do ativo.

Após identificar as ferramentas mais relevantes, o sistema consulta as plataformas de segurança usadas pela Euronext, e os seus resultados são normalizados, deduplicados e agregados numa estrutura comum que permite apresentar uma visão consolidada das vulnerabilidades identificadas.

A Figura 13 representa a arquitetura lógica do *SAS*.

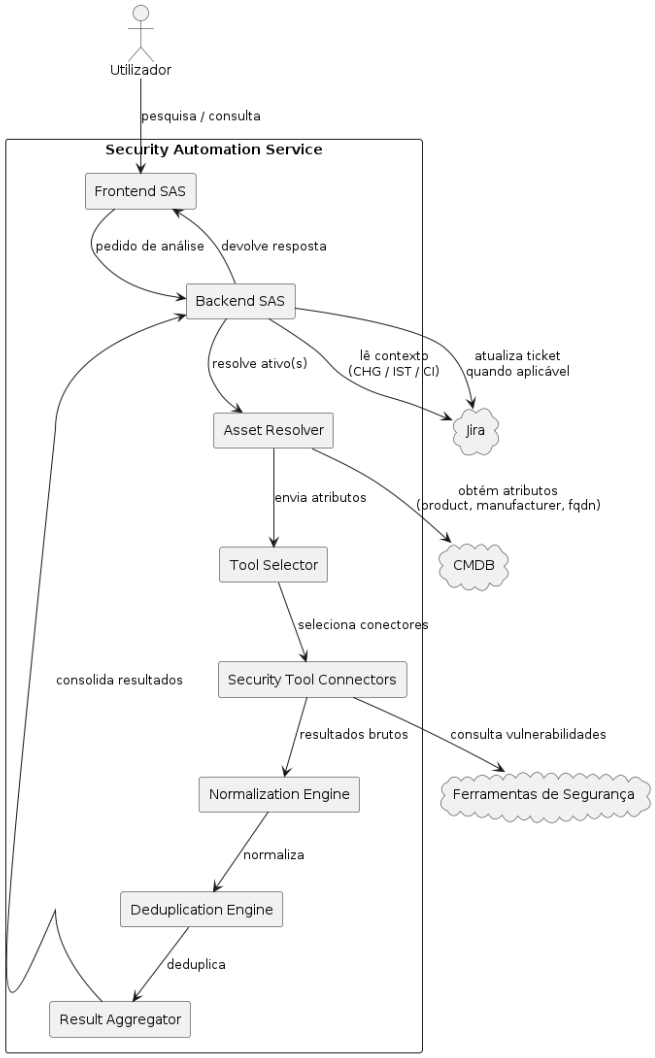


Figura 13 – Arquitetura Lógica do *SAS*

3.3.1.2 Arquitetura Física do SAS

O SAS não mantém a persistência local de dados. Toda a informação relevante é recolhida diretamente das ferramentas de segurança no momento da análise. Para além de não trazer

grande vantagem, pelo menos nesta fase de desenvolvimento, ter uma base de dados dedicada a guardar as pesquisas realizadas, garante que os resultados apresentados refletem sempre o estado mais recente da informação disponível.

A Figura 14 representa a arquitetura física do SAS, ilustrando a distribuição dos componentes da aplicação e as suas integrações com os sistemas externos.

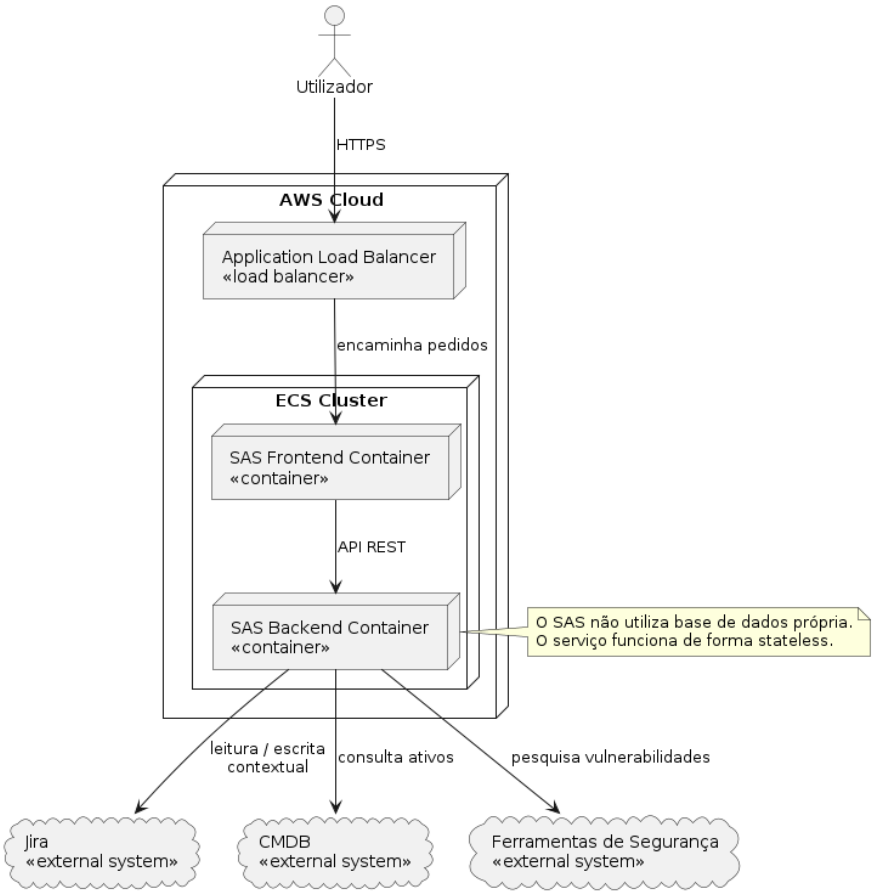


Figura 14 – Arquitetura Física do *SAS*

3.3.2 Arquitetura do RTF

O *RTF* é um serviço que permite organizar ativos e *assessments*, definir prioridades de análise e acompanhar o histórico de avaliações de cada ativo. O sistema fornece mecanismos de rastreabilidade que permitem relacionar as vulnerabilidades identificadas com os respetivos *assessments*.

3.3.2.1 Arquitetura Lógica do RTF

O funcionamento do sistema combina processos automáticos executados pela plataforma com operações iniciadas pelos utilizadores. Esta separação permite automatizar tarefas

operacionais recorrentes, mantendo ao mesmo tempo controlo humano sobre decisões relacionadas com a execução efetiva dos *assessments*.

O sistema executa um processo periódico de sincronização de ativos com a *CMDB* e, durante este processo, são atualizadas informações relevantes sobre os ativos registados. Esta sincronização inclui também o cálculo de sugestões de planeamento de *assessments*, com base nos critérios definidos: criticidade do ativo, histórico de avaliações e vulnerabilidades existentes.

As datas calculadas durante este processo correspondem apenas a sugestões de planeamento e não representam avaliações efetivamente agendadas. Os utilizadores podem consultar a informação dos ativos e rever as sugestões de planeamento apresentadas pelo sistema, podendo alterar as datas sugeridas, alterar prioridades ou criar um *assessment* através da plataforma.

Quando um *assessment* é criado através da plataforma, o sistema gera um *ticket*, do tipo *InfoSec Task* (*IST*). Em determinados casos, o *RTF* pode também invocar o *SAS* para executar automaticamente um *Vulnerability Assessment*.

A Figura 15 representa a arquitetura lógica do *RTF*.

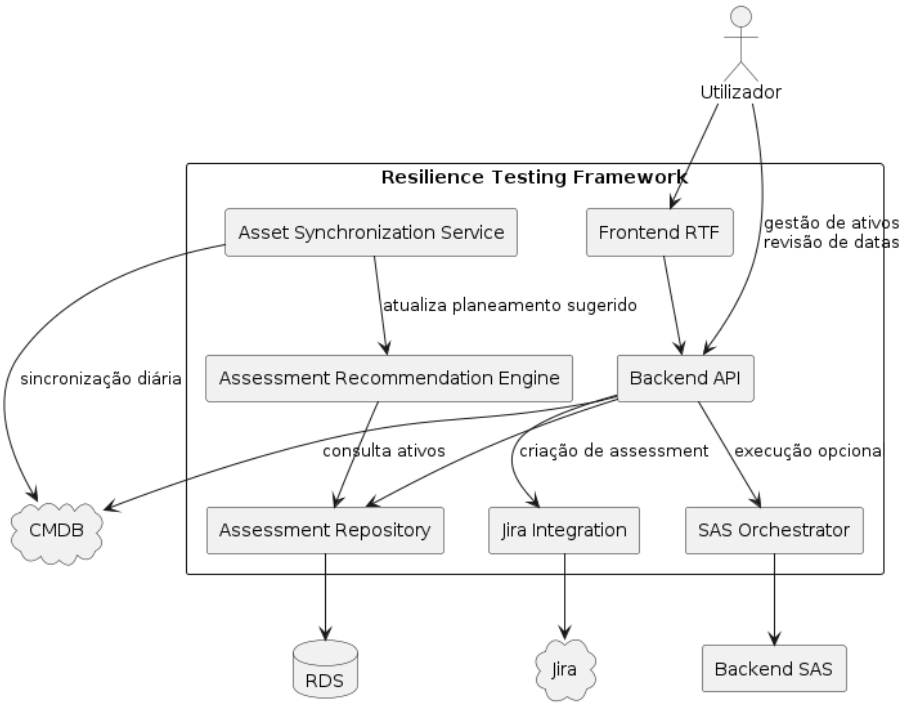


Figura 15 – Arquitetura Lógica do *RTF*

3.3.2.2 Arquitetura Física do RTF

Para suportar a persistência de informação, o *backend* comunica com uma base de dados relacional existente numa *AWS Relational Database Service* (*RDS*), utilizada para armazenar dados associados ao planeamento dos *assessments*, gestão dos ativos e histórico das avaliações.

O *backend* comunica também com sistemas externos utilizados pela Euronext. Estas integrações incluem consultas à *CMDB* para obtenção de informação sobre os ativos, a criação de *tickets* no *Jira* e a invocação do *SAS* quando necessário.

A Figura 16 representa a arquitetura física do *RTF*.

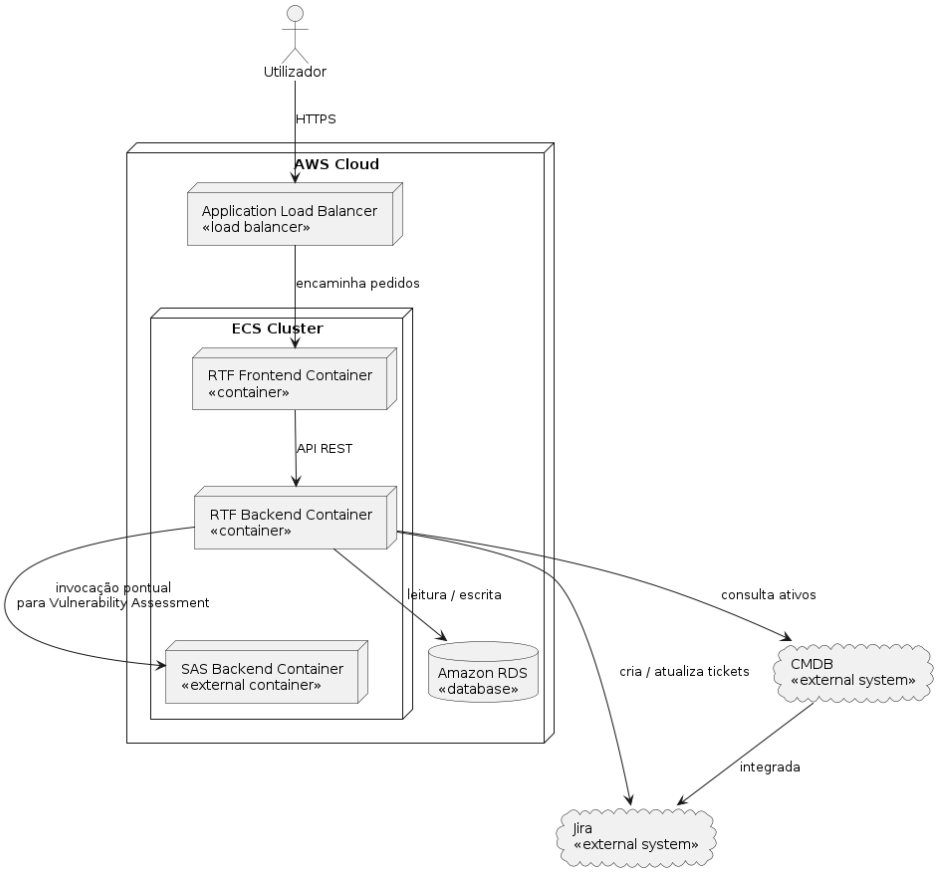


Figura 16 – Arquitetura Física do *RTF*

3.4 Modelo de Dados

O modelo de dados da solução reflete as responsabilidades distintas dos dois componentes principais do *Security Command Hub*. O *SAS* funciona como um serviço de integração responsável pela recolha e consolidação de informação proveniente de diferentes ferramentas de segurança, enquanto o *RTF* mantém persistência de informação associada ao planeamento e execução de *assessments* de segurança.

Esta separação permite reduzir complexidade no *SAS*, que opera como um serviço *stateless*, e concentrar no *RTF* a gestão de informação relacionada com ativos, planeamentos e histórico de *assessments*.

3.4.1 Modelo de Dados do SAS

O *SAS* não possui um modelo de dados persistente próprio. O serviço funciona como um mecanismo de integração responsável por recolher informação de vulnerabilidades associadas a ativos registados na *CMDB* e consolidar essa informação numa estrutura comum durante a execução de cada análise.

Embora o serviço não mantenha persistência local de dados, é necessário definir uma estrutura lógica que represente a informação manipulada durante o processo de análise. Esta estrutura, representada na Figura 17, inclui entidades conceptuais como ativos, vulnerabilidades, ferramentas de segurança e resultados agregados.

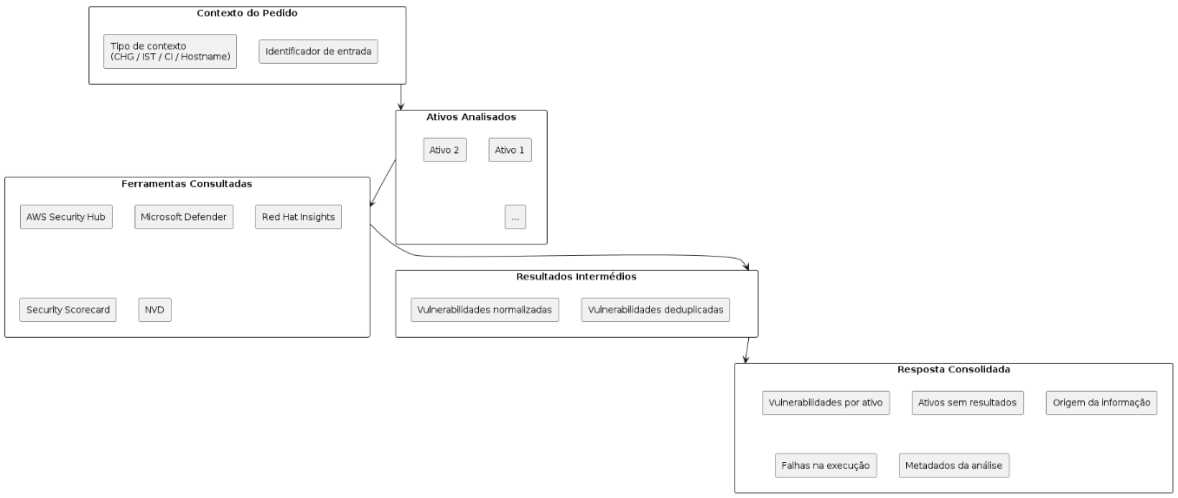


Figura 17 – Estrutura conceptual da resposta à análise de vulnerabilidades

3.4.2 Modelo de Dados do RTF

Já o *RTF* mantém persistência de informação associada ao planeamento e execução de *assessments*. Este componente utiliza uma base de dados relacional para armazenar informação sobre ativos, avaliações de segurança e histórico.

A Figura 18 representa o modelo de dados do *RTF*.

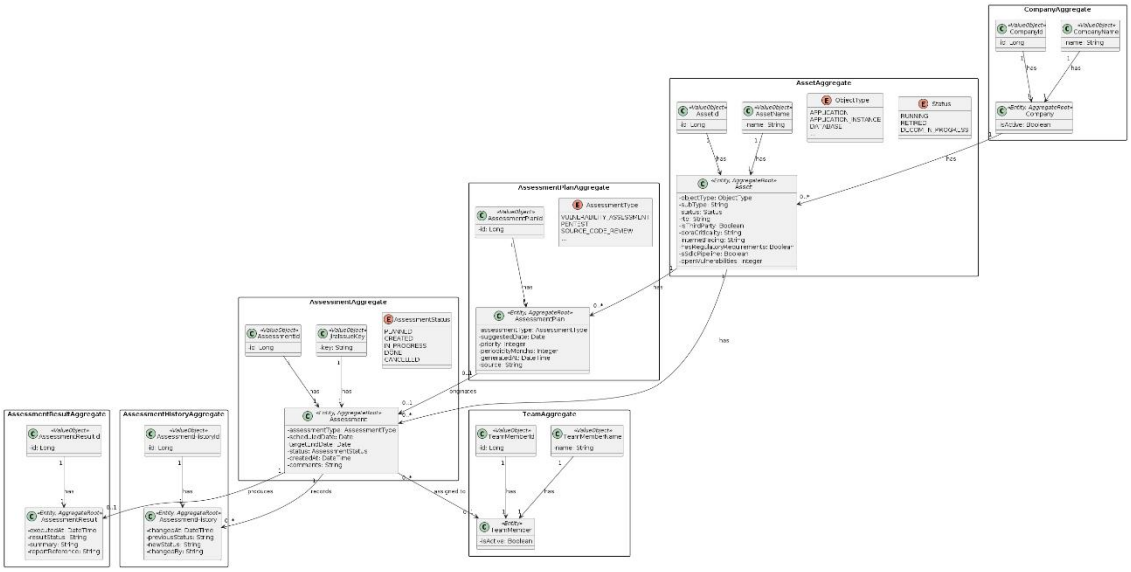


Figura 18 – Modelo de Dados do *RTF*

3.5 Fluxos Operacionais

Este subcapítulo descreve os principais fluxos operacionais suportados pela solução proposta. Estes fluxos permitem compreender de que forma os componentes do *Security Command Hub* colaboram para apoiar a gestão de vulnerabilidades e o planeamento de *assessments*.

Os fluxos apresentados refletem cenários centrais no funcionamento da solução, incluindo a pesquisa de vulnerabilidades associadas a ativos específicos, a execução de *Vulnerability Assessments*, a sincronização periódica de ativos com a *CMDB* e a interação dos utilizadores com o processo de planeamento de avaliações de segurança.

3.5.1 Pesquisa de Vulnerabilidades por Ativo

Este fluxo pode ser iniciado de forma independente através do *SAS*, permitindo obter uma visão consolidada das vulnerabilidades associadas a um determinado elemento do inventário corporativo.

O processo inicia-se quando o utilizador submete um pedido de análise associado a um ativo ou a um *ticket* proveniente do *Jira* (no caso das *Change Requests* ou *InfoSec Tasks*). A partir desse pedido, o sistema identifica o ativo relevante (que no caso de um *ticket*, pode existir mais do que um ativo), e consulta a *CMDB* para obter os atributos necessários dos ativos identificados para efetuar a pesquisa.

Com base nesses atributos, o sistema determina quais as ferramentas de segurança relevantes para o ativo em análise. O serviço executa então consultas às *APIs* das plataformas

correspondentes, recolhendo vulnerabilidades associadas ao produto, versão, fabricante ou outro identificador disponível.

Os resultados obtidos são depois normalizados, deduplicados e agregados numa estrutura comum que permite apresentar uma visão consolidada das vulnerabilidades identificadas.

A Figura 19 representa este fluxo operacional num diagrama dedicado.

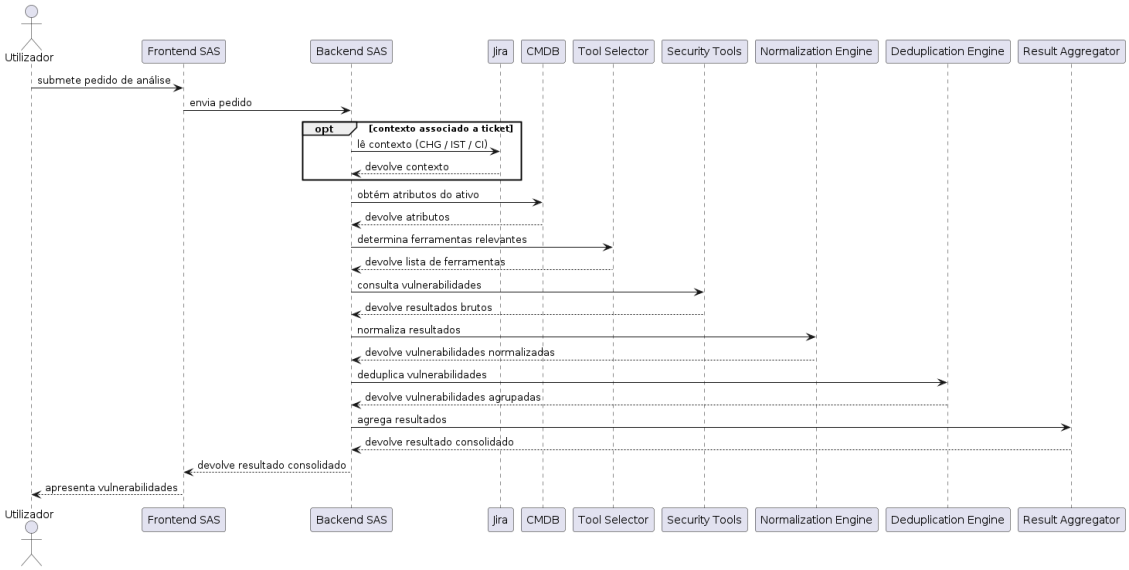


Figura 19 – Diagrama de sequência da pesquisa de vulnerabilidades por ativo

3.5.2 Criação e Execução de um Vulnerability Assessment

Outro fluxo relevante corresponde à criação e execução de um *assessment* do tipo *Vulnerability Assessment*. Este fluxo evidencia a interação pontual entre o *RTF* e o *SAS*.

O processo inicia-se quando o utilizador decide criar um *assessment* para o ativo registado na plataforma. O *RTF* recolhe a informação necessária sobre o ativo, associando-lhe o tipo de avaliação definido e a data de execução pretendida.

Após isto, o sistema gera o *ticket* correspondente no *Jira*. Quando o *assessment* criado corresponde a um *Vulnerability Assessment*, o *RTF* pode desencadear a execução do *SAS* para recolher automaticamente vulnerabilidades associadas ao ativo.

Neste cenário, o *SAS* executa o fluxo de pesquisa de vulnerabilidades descrito acima e devolve os resultados ao *RTF*. O sistema regista então a evidência recolhida no *ticket* criado para o *assessment*, fechando o mesmo.

A Figura 20 representa este fluxo operacional num diagrama dedicado.

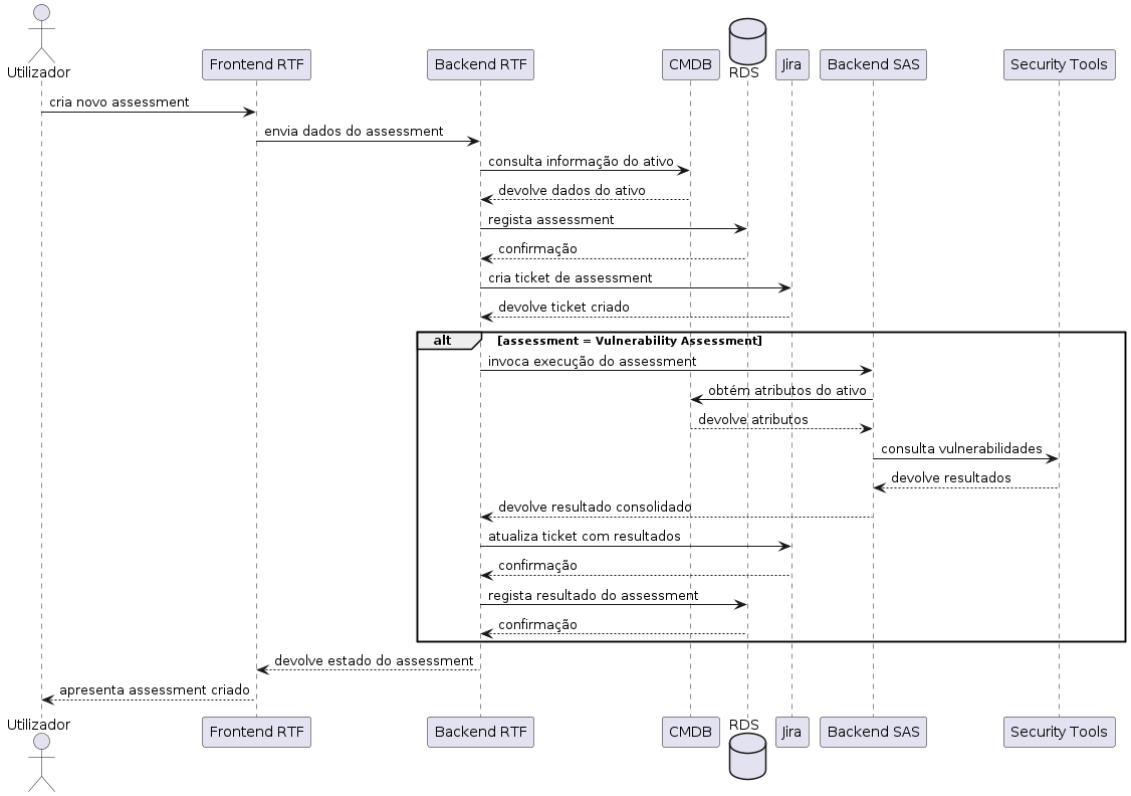


Figura 20 – Diagrama de sequência da criação e execução de um *Vulnerability Assessment*

3.5.3 Sincronização Diária de Ativos e Sugestões de Planeamento

O *RTF* suporta também um fluxo automático de sincronização de ativos com a *CMDB*. Este fluxo tem como objetivo manter atualizada a informação sobre o inventário corporativo e apoiar o cálculo de sugestões de planeamento de *assessments*.

Durante a execução deste processo, o sistema consulta a *CMDB* e atualiza localmente a informação relevante sobre os ativos registados no inventário. Esta atualização inclui atributos necessários para o planeamento de *assessments*, como criticidade, exposição, estado atual do ativo e outros metadados relevantes.

Com base nesta informação, o sistema executa regras de planeamento que permitem calcular sugestões de avaliação, como o tipo de teste e as datas em que este teste deverá ser executado.

As datas geradas durante este processo correspondem apenas a sugestões de planeamento e não resultam automaticamente na criação de *assessments*. O sistema armazena essas sugestões para posterior revisão.

Este fluxo permite automatizar tarefas operacionais recorrentes e garantir que o planeamento de *assessments* parte de informação atualizada sobre o inventário corporativo.

A Figura 21 representa este fluxo operacional num diagrama dedicado.

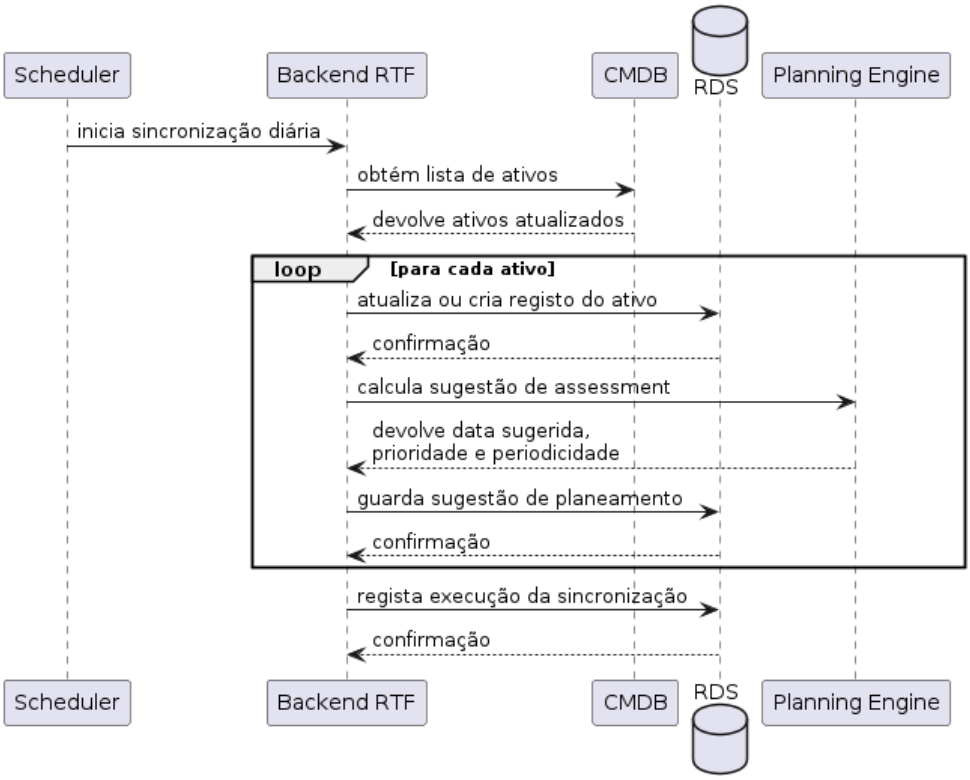


Figura 21 – Diagrama de sequência da sincronização de ativos e sugestões de planeamento

3.5.4 Revisão e Criação de Assessments pelo Utilizador

Embora o sistema calcule sugestões de planeamento de forma automática, a criação efetiva de *assessments* permanece sob o controlo do utilizador. Este fluxo permite combinar automação com validação humana no processo de planeamento de avaliações de segurança.

O processo inicia-se quando o utilizador consulta os ativos e as sugestões de planeamento apresentadas pela plataforma. Com base nessa informação, o utilizador pode rever as datas sugeridas, ajustar prioridades ou selecionar o tipo de *assessment* a criar.

Quando o utilizador confirma a criação de um novo *assessment*, o sistema regista o mesmo e gera o *ticket* associado no *Jira*. A partir desse momento, o *assessment* passa a fazer parte do histórico do ativo e pode ser acompanhado através da plataforma.

A Figura 22 representa este fluxo operacional num diagrama dedicado.

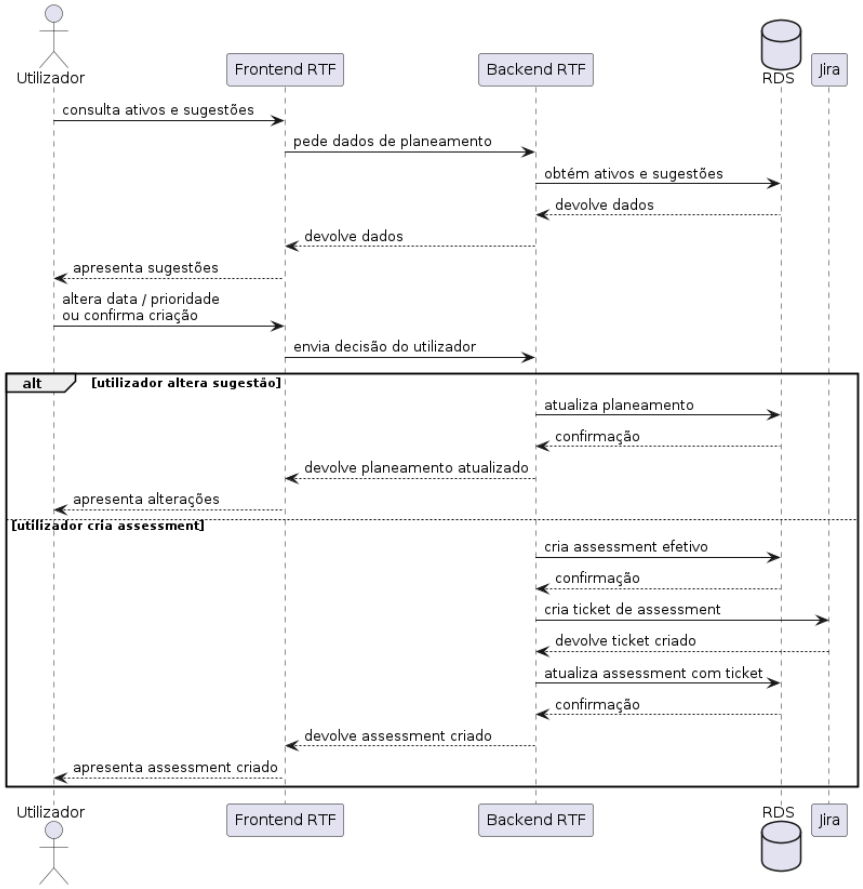


Figura 22 – Diagrama de sequência da revisão e criação de *assessments*

3.6 Algoritmos

Este subcapítulo descreve os algoritmos utilizados pelo *RTF* para apoiar o processo de planeamento de *assessments*.

Estes algoritmos traduzem os critérios definidos em regras operacionais na arquitetura da solução, permitindo automatizar a priorização de ativos, a identificação do tipo de teste mais adequado e o cálculo das sugestões de planeamento.

Para facilitar a interpretação da análise da complexidade apresentada nos subcapítulos seguintes, a Tabela 8 resume o significado das principais notações utilizadas:

Tabela 8 – Notação Big O (incompleta) para interpretar a análise de complexidade

Notação	Designação	Interpretação
O(1)	Complexidade constante	O número de operações não depende do tamanho da tarefa. O custo mantém-se estável para cada execução.

O(log n)	Complexidade logarítmica	O número de operações cresce lentamente à medida que o tamanho da entrada aumenta.
O(n)	Complexidade linear	O número de operações cresce de forma proporcional ao tamanho da entrada.
O(n log n)	Complexidade linear-logarítmica	O custo cresce mais do que linearmente.
$O(n^2)$	Complexidade quadrática	O número de operações cresce rapidamente com o tamanho da entrada.

Neste contexto, a análise da complexidade incide apenas sobre a lógica interna dos algoritmos, assumindo que os dados de entrada já se encontram disponíveis no momento de execução.

3.6.1 Algoritmo de Priorização de Assets

O algoritmo de priorização de *assets* determina a categoria de prioridade atribuída a cada ativo com base nos atributos obtidos a partir da *CMDB* e de informação complementar associada ao contexto do ativo. O objetivo do algoritmo consiste em ordenar os ativos segundo critérios de criticidade operacional e exposição, permitindo suportar o planeamento de *assessments* de forma consistente.

As entradas do algoritmo incluem o tipo de ativo, o valor de *RTO*, a existência de requisitos regulamentares, a criticidade definida no contexto do *DORA*, se o ativo é *legacy* e o nível de exposição à internet. Estes atributos representam o conjunto mínimo de informação necessária para classificar o ativo segundo as regras definidas pela Euronext.

A saída do algoritmo corresponde a uma categoria discreta de prioridade, representada por um nível entre **#0** e **#6**. Esta categoria é depois utilizada pelo *RTF* como base para apoiar o processo de planeamento e cálculo de periodicidade de *assessments*.

O algoritmo segue uma abordagem determinística baseada em regras sequenciais. A primeira regra verifica se o ativo apresenta um *RTO* inferior ou igual a duas horas. Quando esta condição se verifica, o sistema atribui imediatamente a prioridade ***#0 – RTO <2***, por se tratar do nível de prioridade mais elevado.

Quando o algoritmo não satisfaz esta condição, o algoritmo distingue entre ativos do tipo *Application & Components* e os restantes tipos. Para ativos do tipo *Application & Components*, o sistema verifica primeiro a existência de requisitos regulamentares. Quando estes requisitos estão presentes, o ativo recebe a prioridade ***#1 – Regulatory Requirements***.

Na ausência de requisitos regulamentares, o algoritmo avalia a criticidade *DORA*. Se o ativo estiver classificado como *Critical Function*, o sistema atribui ***#2 – Dora Critical Function***, exceto quando o ativo se encontra marcado como *legacy*, caso em que a prioridade passa para ***#3 – Dora Legacy***. Se o ativo estiver classificado como *Important Function*, o sistema atribui ***#4 – Dora Important Function***, exceto quando o ativo é *legacy*, situação em que a prioridade volta a ser ***#3 – Dora Legacy***.

Quando nenhuma destas condições se aplica, o sistema utiliza a exposição à internet como critério final. Ativos com exposição *Internet Facing* ou *Both* recebem ***#5 – Internet Risk Driven***. Os restantes ativos recebem ***#6 – Others Risk Driven***.

Para ativos que não pertencem ao tipo *Application & Components*, o algoritmo aplica a mesma lógica relativa à criticidade DORA, ao estado de legacy e à exposição à internet, omitindo apenas a verificação específica de requisitos regulamentares associada ao ramo das aplicações.

A complexidade temporal do algoritmo é O(1), porque a decisão resulta de um conjunto fixo de verificações condicionais aplicadas a cada ativo. A complexidade espacial é também O(1), dado que o algoritmo não utiliza estruturas auxiliares dependentes do tamanho da entrada.

A Figura 23 apresenta o diagrama de decisão associado ao algoritmo de priorização, permitindo visualizar de forma estruturada o fluxo de avaliação e as condições consideradas.

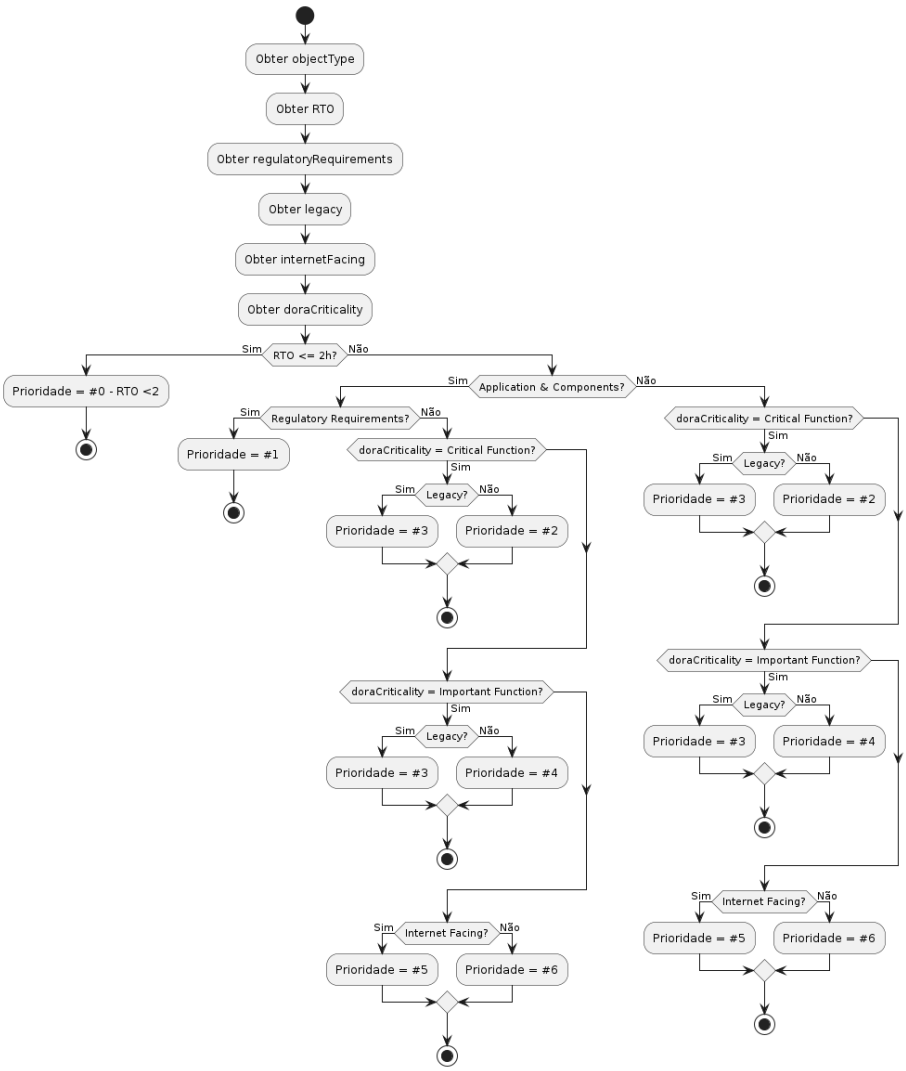


Figura 23 – Diagrama de Decisão do Algoritmo de Priorização de *Assets*

3.6.2 Algoritmo de Tipo de Teste

O algoritmo de tipo de teste determina qual a avaliação de segurança mais adequada para cada ativo com base em atributos obtidos através da *CMDB* e informação complementar associada ao ativo. O objetivo do algoritmo consiste em selecionar o tipo de avaliação que melhor se adequa ao risco, natureza e histórico do ativo.

As entradas do algoritmo incluem o tipo de ativo, se é *legacy*, a natureza *third-party* da aplicação, o tipo de produto associado fornecedor, o valor de *RTO*, a existência de requisitos regulamentares, a exposição à internet, a presença na pipeline *S-SDLC*, o número de vulnerabilidades abertas e a data do último teste realizado.

A saída do algoritmo corresponde ao tipo de teste recomendado para o ativo. Entre os valores possíveis estão: Risk Assessment, Manual Penetration Testing, Automated Penetration Testing, Third Party Assessment (TPA), Network Security Assessment / Vulnerability Assessment and scans, Vulnerability Assessment and scans e Physical Security Review.

O algoritmo começa por verificar se o ativo pertence à categoria *Application & Components*. Quando esta condição se verifica, o sistema aplica um conjunto de regras específicas para aplicações. Caso contrário, o sistema utiliza um conjunto de regras simplificadas baseadas no tipo de objeto.

No caso das aplicações, a primeira regra verifica se o ativo está marcado como *legacy*. Quando esta condição se verifica, o sistema atribui imediatamente o tipo ***Risk Assessment***. Se o ativo não for *legacy*, o algoritmo avalia o tipo de produto do fornecedor. Quando o produto é classificado como *Customized* ou *Developed / Customized*, o sistema seleciona ***Manual Penetration Testing***.

Quando esta condição não se verifica, o algoritmo analisa se a aplicação é *third-party* ou se o produto é do tipo *Off-the-shelf*. Nestes casos, o sistema seleciona ***Third Party Assessment*** (***TPA***).

Se nenhuma destas condições se aplicar, o algoritmo verifica se o ativo apresenta *RTO* <=2 ou se possui requisitos regulamentares. Quando uma destas condições se verifica, a decisão depende da exposição à internet. Ativos expostos à internet recebem ***Manual Penetration Testing,*** enquanto ativos internos recebem ***Automated Penetration Testing***.

Quando não existe *RTO* crítico nem requisitos regulamentares, o sistema verifica se o ativo está presente na pipeline *S-SDLC*. Se estiver, o algoritmo analisa o número de vulnerabilidades abertas. Quando existem vulnerabilidades abertas, o tipo de teste selecionado é ***Automated Penetration Testing***. Quando não existem vulnerabilidades abertas, o sistema utiliza o histórico do último teste e a exposição à internet para distinguir entre ***Manual Penetration Testing*** e ***Automated Penetration Testing***.

Quando a aplicação não está abrangida pela pipeline *S-SDLC*, o algoritmo aplica a mesma lógica baseada no histórico do último teste e na exposição à internet, mantendo ***Automated***

Penetration Testing como escolha por defeito para ativos recentemente avaliados e não expostos externamente.

Para ativos que não pertencem à categoria *Application & Components*, o algoritmo aplica regras baseadas no tipo de objeto. Ativos classificados como *Software*, *Hardware* ou *AWS* recebem ***Network Security Assessment / Vulnerability Assessment and scans***. Ativos classificados como *Azure* ou *Openshift* recebem ***Vulnerability Assessment and scans***. Ativos físicos recebem ***Physical Security Reviews***. No caso de ativos do tipo *Service*, o sistema distingue se são *third-party*. Quando essa condição se verifica, o tipo atribuído é ***Manual Penetration Testing***. Nos restantes casos, o sistema devolve um erro de classificação.

A complexidade temporal do algoritmo é O(1), porque a decisão depende apenas de um conjunto fixo de verificações condicionais para cada ativo. A complexidade espacial é também O(1), dado que não são utilizadas estruturas auxiliares dependentes do tamanho da entrada.

A Figura 24 e a Figura 25 apresentam o diagrama de decisão associado ao algoritmo de tipo de teste, permitindo visualizar de forma estruturada as condições consideradas durante o processo de seleção.

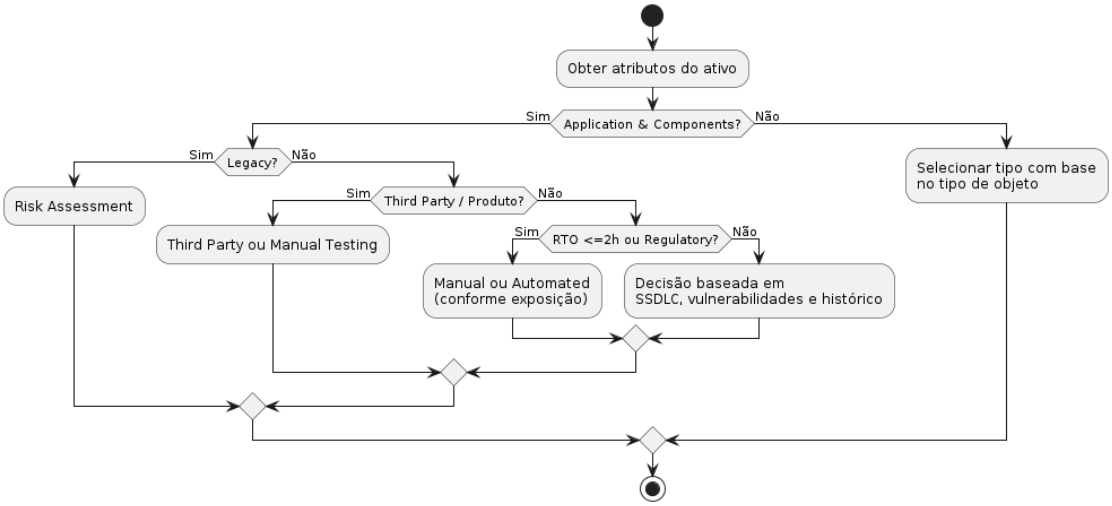


Figura 24 – Diagrama de Decisão do Algoritmo de Tipo de Teste Generalizado

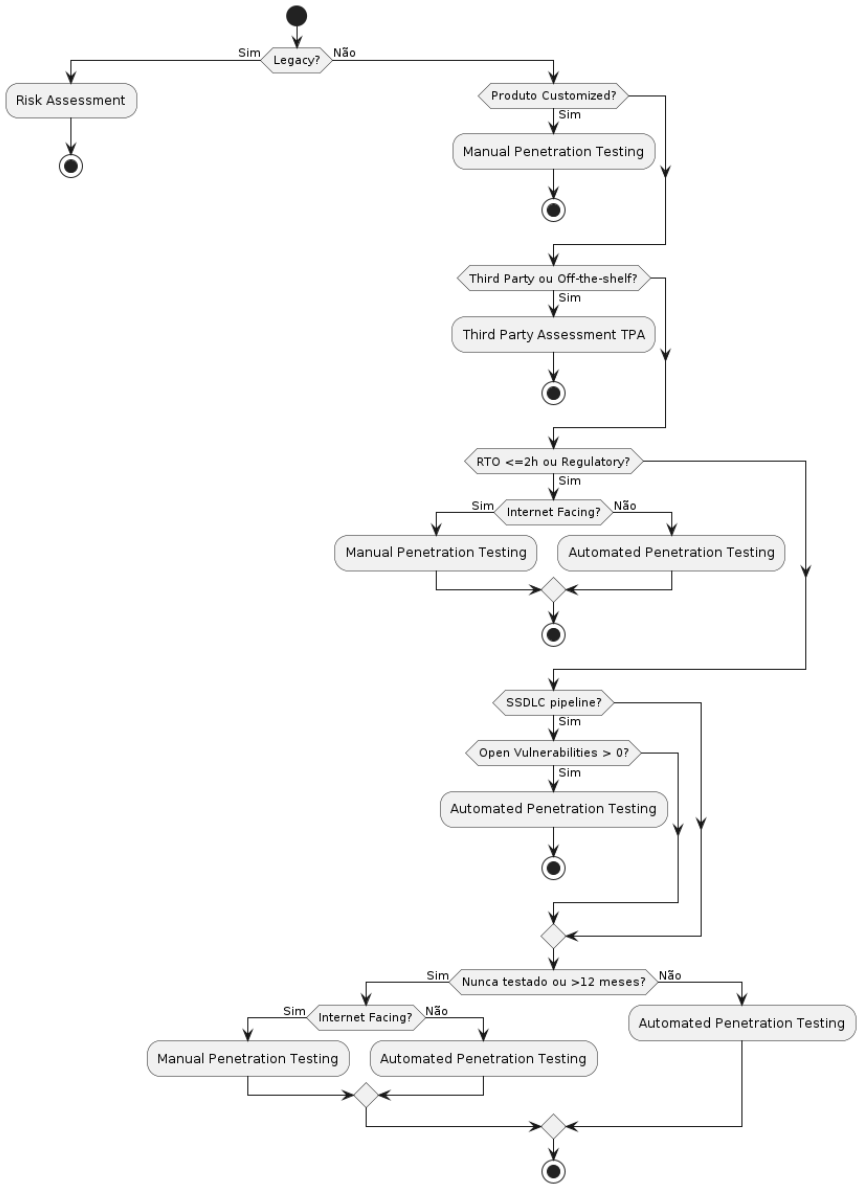


Figura 25 – Diagrama de Decisão do Algoritmo de Tipo de Teste Detalhado para *Application & Components*

3.6.3 Algoritmo de Agendamento de Assessments

O algoritmo de agendamento de *assessments* determina a data sugerida para a próxima avaliação de segurança de cada ativo. O objetivo do algoritmo consiste em combinar regras automáticas com mecanismos de controlo manual, permitindo adaptar o planeamento às decisões do utilizador sem comprometer a consistência temporal do ciclo de avaliações.

As entradas do algoritmo incluem a data do último teste realizado, a data de início associada a esse teste, a criticidade definida no contexto do *DORA*, a prioridade calculada previamente pelo

sistema, o intervalo de periodicidade definido para o tipo de teste, a eventual data manual definida pelo utilizador, a data efetiva registada no ativo e o indicador *To be Delayed* (*TBD*).

A saída do algoritmo corresponde a três valores: a data calculada automaticamente pelo sistema, a data manual eventualmente mantida após validação e a data final formatada para apresentação ou utilização operacional.

O algoritmo inicia a execução verificando a existência do histórico de testes. Quando não existe qualquer avaliação anterior, o sistema gera uma data futura distribuída ao longo do período seguinte, permitindo introduzir o ativo no ciclo de planeamento sem exigir histórico prévio.

Quando existe histórico, o algoritmo valida primeiro a consistência da data efetiva. Se esta data se encontrar desatualizada em relação ao último teste realizado, o sistema invalida-a e força o recálculo automático da próxima avaliação.

O algoritmo verifica depois a existência de uma data manual definida pelo utilizador. Quando essa data ainda é válida, isto é, quando não foi ultrapassada pelo teste mais recente, o sistema mantém essa definição e termina o cálculo sem nova sugestão automática. Quando a data manual já foi ultrapassada, o sistema remove essa referência e continua o processamento.

Em seguida, o algoritmo trata o caso específico em que o indicador *TBD* está ativo. Quando esta condição se verifica e o último teste foi realizado no ano corrente, o sistema calcula a próxima data com base no último teste ou na sua data de início, adicionando o intervalo de periodicidade correspondente.

O algoritmo inclui também um caso especial para ativos com periodicidade anual, histórico anterior ao ano corrente e data efetiva projetada para um ano futuro. Nestas situações, o sistema gera uma data distribuída no ano atual, permitindo evitar um desalinhamento excessivo entre planeamento e execução.

Quando não existe data efetiva válida, o sistema calcula a próxima avaliação a partir da data do último teste, adicionando o intervalo de periodicidade definido. Se a data calculada se situar no ano corrente ou num período já ultrapassado, o sistema ajusta a sugestão para o ano seguinte.

A complexidade do algoritmo é O(1), porque a decisão resulta de um número fixo de verificações condicionais e operações de cálculo sobre datas. A complexidade espacial é também O(1), dado que o algoritmo opera apenas sobre um conjunto constante de variáveis de entrada e saída.

A Figura 26 apresenta uma visão geral do algoritmo de agendamento, destacando as principais decisões associadas ao histórico de testes, às definições manuais e aos mecanismos automáticos.

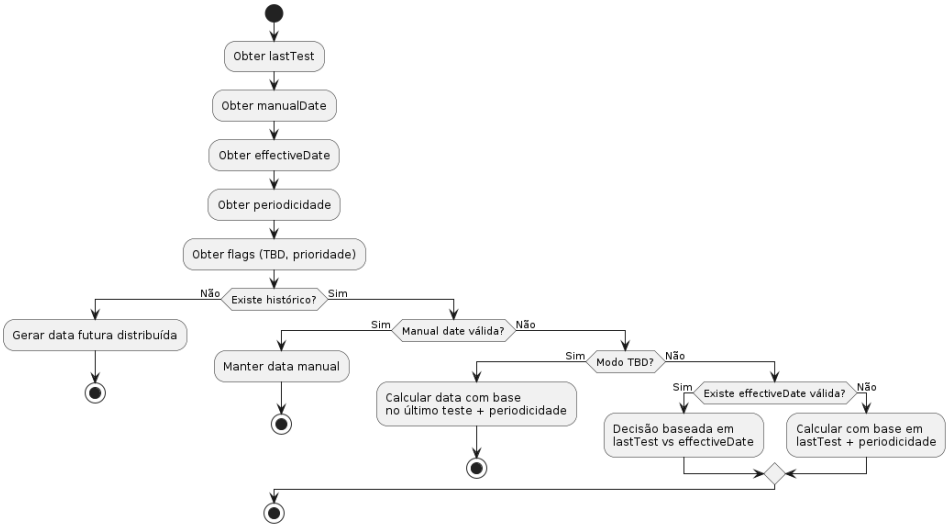


Figura 26 – Diagrama de Decisão do Algoritmo de Agendamento de *Assessments* Generalizado

A Figura 27 detalha o processo de cálculo da data sugerida, incluindo a validação das datas existentes, tratamento de exceções e aplicando a periodicidade definida.

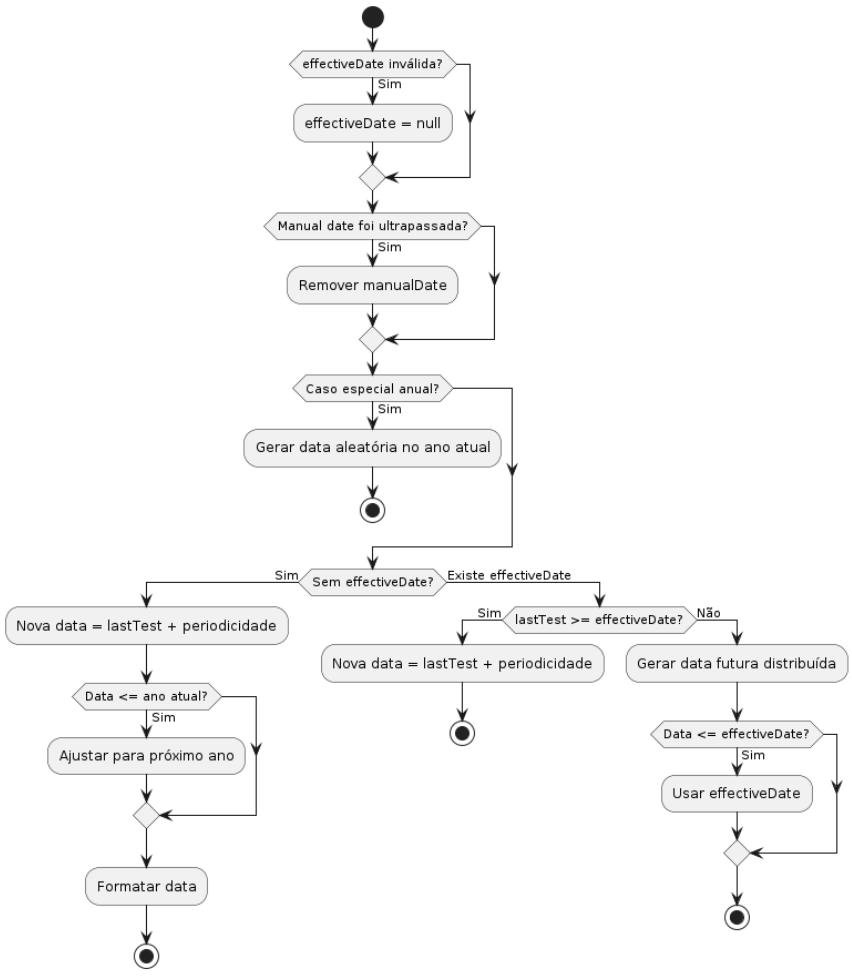


Figura 27 – Diagrama de Decisão do Algoritmo de Agendamento de *Assessments* Detalhado

3.7 Critérios de Avaliação

O presente subcapítulo descreve a forma como a solução proposta suporta a avaliação definida previamente, estabelecendo a ponte entre as métricas e as questões de investigação identificadas e os mecanismos concretos implementados no *Security Command Hub*.

A estratégia de avaliação baseia-se na comparação entre o processo manual atualmente utilizado e a abordagem automatizada proposta. As métricas apresentadas previamente são suportadas por dados recolhidos diretamente dos componentes do sistema, permitindo avaliar de forma objetiva o impacto da solução.

O *SAS* suporta as métricas relacionadas com a consolidação e a normalização dos dados, enquanto as métricas relacionadas com o planeamento são suportadas pelo *RTF*.

Para além das métricas definidas, foram estabelecidos critérios mínimos de aceitação que permitem validar se a solução implementada cumpre os requisitos essenciais para suportar o processo de gestão de vulnerabilidades e planeamento. Estes critérios não substituem as métricas de avaliação, mas definem um nível mínimo de funcionamento necessário para considerar a solução viável no contexto organizacional.

A Tabela 9 apresenta os critérios mínimos de aceitação da solução e o método de verificação associado aos mesmos.

Tabela 9 – Critérios Mínimos de Aceitação da Solução

Critério	Descrição	Verificação
Consolidação de Vulnerabilidades	O sistema agrega vulnerabilidades de múltiplas fontes e associa-as a ativos da *CMDB*	Execução de cenários de teste e validação dos resultados consolidados
Integração com *CMDB*	O sistema sincroniza ativos e atributos com a *CMDB* de forma consistente	Comparação entre dados do sistema e registos da *CMDB*
Cálculo de prioridade	O sistema classifica ativos de acordo com as regras definidas no algoritmo de priorização	Execução de testes com diferentes combinações de atributos
Determinação de tipo de teste	O sistema seleciona o tipo de *assessment* com base nas características do ativo	Validação dos resultados face às regras definidas no algoritmo
Sugestão de planeamento	O sistema gera datas sugeridas de *assessments* com base em periodicidade e histórico	Análise das datas calculadas em diferentes cenários
Criação de *assessments*	O utilizador consegue criar *assessments* e associá-los a ativos	Testes funcionais e validação da persistência no sistema

Integração com *Jira*	O sistema cria e atualiza *tickets* associados aos *assessments*	Verificação da criação e atualização de *tickets* reais
Execução automática de *assessments*	O sistema invoca o *SAS* para execução de *Vulnerability Assessments*	Execução de cenários *end-to-end* e validação dos resultados
Persistência e histórico	O sistema mantém histórico completo de *assessments* e resultados	Consulta direta à base de dados e validação dos registos
Rastreabilidade	O sistema permite relacionar ativos, vulnerabilidades e *assessments*	Validação através da navegação e análise das ligações entre entidades

3.8 Decisões Arquiteturais e Trade-offs

Estas decisões, validadas com os *stakeholders* e membros da equipa de segurança da Euronext, resultam da necessidade de equilibrar os requisitos funcionais, restrições técnicas e objetivos de integração com os sistemas existentes na organização.

Uma das decisões centrais consiste na separação da solução em dois módulos. Esta separação permite isolar responsabilidades, promovendo o desacoplamento entre componentes e facilitando a evolução independente de cada módulo. No entanto, esta abordagem introduz complexidade adicional na integração entre serviços e na gestão de comunicação entre os componentes.

A decisão de não substituir as ferramentas de segurança existentes, mas sim integrar com estas através de *APIs*, permite preservar os investimentos já realizados pela Euronext e reduzir o impacto da adoção da solução. Esta abordagem garante compatibilidade com o ecossistema existente, mas implica lidar com a heterogeneidade de interfaces, formatos de dados e limitações específicas de cada ferramenta.

A integração da *CMDB* através de *APIs* constitui outra decisão fundamental. Esta abordagem permite a sincronização bidirecional de dados e garante que o inventário de ativos se mantém consistente entre sistemas. No entanto, esta dependência introduz acoplamento com a qualidade e disponibilidade da *CMDB*, podendo afetar o comportamento do sistema em cenários de dados incompletos ou inconsistentes.

A utilização de algoritmos para a automatização de decisões ao nível do processo de agendamento permite garantir previsibilidade e transparência. Facilita a validação e auditoria dos resultados, mas limita a capacidade de adaptação futura a padrões mais complexos que poderiam ser explorados com modelos de *machine learning*.

A arquitetura baseada em serviços containerizados executados em ambiente *cloud* permite escalar componentes de forma independente e garantir isolamento entre módulos. Esta

abordagem suporta requisitos de escalabilidade e disponibilidade, mas implica custos operacionais adicionais e necessidade de gestão de infraestrutura.

A persistência centralizada no *RTF* permite manter um histórico consistente de ativos, *assessments* e suportar a análise de dados. No entanto, esta decisão cria dependência de uma única base de dados para o planeamento, exigindo mecanismos adequados de disponibilidade e recuperação.

A sincronização periódica entre sistemas introduz também um compromisso entre desempenho e atualidade dos dados. Embora esta abordagem reduza o custo de consultas contínuas e melhore a eficiência operacional, pode existir latência entre o estado real dos ativos ou vulnerabilidades nas plataformas externas e a informação usada internamente pelo sistema no momento do planeamento.

A dependência de *APIs* externas implica exposição a limitações que não são controladas pela solução, incluindo indisponibilidade temporária, *rate limits*, alterações contratuais e inconsistências nas respostas devolvidas. A implementação inclui mecanismos de *retry* e tratamento de falhas transitórias, mas estes mecanismos não eliminam totalmente o risco associado à variabilidade das integrações externas.

Estas decisões refletem um compromisso entre automação, controlo, integração e complexidade. A solução proposta privilegia uma integração com os sistemas existentes, transparência nos processos de decisão e suporte ao utilizador, em detrimento de uma solução 100% autónoma. Este equilíbrio permite alinhar a solução com o contexto organizacional da Euronext e com os requisitos de rastreabilidade e conformidade definidos pelo *DORA*, embora mantenha limitações inerentes à qualidade dos dados disponíveis, ao comportamento das integrações externas e ao custo operacional da arquitetura adotada.

4 Implementação

O presente capítulo descreve a implementação do *Security Command Hub*, detalhando as decisões técnicas adotadas, as tecnologias utilizadas e os mecanismos desenvolvidos para suportar os fluxos operacionais e algoritmos definidos no capítulo anterior. Este capítulo apresenta a concretização da solução proposta, evidenciando como a arquitetura desenhada foi materializada num sistema funcional.

4.1 Metodologia

A implementação da solução segue uma abordagem orientada a serviços, baseada na separação entre componentes responsáveis pela consolidação de vulnerabilidades e pelo planeamento de *assessments*.

O desenvolvimento foi realizado de forma incremental, permitindo validar progressivamente integrações, fluxos operacionais e algoritmos centrais da solução. Esta abordagem permitiu verificar o comportamento de cada componente antes da sua articulação com os restantes serviços do sistema.

Os serviços foram desenvolvidos como aplicações independentes, encapsuladas em *containers* e executadas em ambiente *cloud*. Esta estratégia permite isolamento entre componentes e controlo sobre o ciclo de execução de cada serviço.

4.2 Tecnologias

A seleção das tecnologias utilizadas na implementação da solução foi realizada juntamente com a equipa de segurança da Euronext, tendo em consideração não apenas os requisitos técnicos da solução, mas também aspetos relacionados com adoção futura, manutenção e familiaridade da equipa com a *stack* escolhida.

A implementação dos serviços de *backend* do *SAS* e do *RTF* foi realizada em *Python*, utilizando a *framework FastAPI*. O uso de *Python* adequa-se ao tipo de processamento necessário na solução. Já a utilização de *FastAPI* permite estruturar os serviços de *backend* em torno de *APIs* bem definidas, facilitando a comunicação entre componentes e a integração com sistemas externos. Esta *framework* suporta validação de dados, definição clara de *endpoints* e geração automática de documentação técnica.

As interfaces de *frontend* das duas aplicações foram desenvolvidas em *React*. A escolha desta tecnologia permite separar a camada de apresentação da lógica de negócio e construir interfaces modulares orientadas a componentes, facilitando a organização da aplicação e evolução independente das aplicações.

A execução local e simulações reais da solução foram concebidas com recurso ao *Docker*, permitindo encapsular os *frontend's* e os *backend's* de cada aplicação em *containers* independentes.

Por fim, para a persistência de dados, neste caso apenas do *RTF*, foi escolhido o uso de *MySQL*. A escolha desta tecnologia resulta da necessidade de manter a informação estruturada num modelo relacional consistente, adequado aos requisitos de rastreabilidade e consulta suportados pela solução.

4.3 Consolidação Multi-Fonte de Vulnerabilidades

A implementação desta funcionalidade foi realizada no *SAS*, que atua como camada de integração entre o *Jira*, a *CMDB* e as ferramentas de segurança utilizadas pela organização.

O ponto de entrada principal do serviço recebe pedidos de pesquisa associados a diferentes tipos de contexto, nomeadamente *changes*, *infosec tasks* ou ativos. A implementação começa por interpretar esse contexto e determinar se o pedido se refere diretamente a um ativo ou se exige resolução prévia através do *Jira* e da *CMDB*.

Quando o contexto recebido não corresponde diretamente a um ativo, o *SAS* consulta o *Jira* para identificar os elementos associados ao pedido. Esta etapa permite recuperar informação suficiente para localizar os ativos relevantes na *CMDB* e iniciar a pesquisa de vulnerabilidades a partir desses registos.

Após a identificação dos ativos, o serviço obtém atributos relevantes da *CMDB*, incluindo campos como produto, fabricante, nome de domínio, entre outros. Estes atributos suportam a classificação dos ativos e permitem determinar quais os conectores que devem ser utilizados para consultar as ferramentas de segurança adequadas a cada caso.

A integração com as fontes de vulnerabilidades foi implementada através de conectores especializados, cada um responsável pela comunicação com uma ferramenta ou plataforma específica. Desta forma, conseguimos encapsular detalhes de autenticação, construção de

pedidos e tratamento das respostas, reduzindo o acoplamento entre a lógica central do *SAS* e as particularidades de cada ferramenta.

Depois da seleção dos conectores adequados, o serviço executa as consultas necessárias e recolhe os resultados devolvidos por cada ferramenta. Como cada fonte apresenta formatos, campos e classificações diferentes, a implementação inclui uma fase de transformação que converte os resultados obtidos para uma estrutura interna comum.

Esta normalização permite representar vulnerabilidades de forma consistente, independentemente da origem dos dados. A transformação inclui a normalização de campos como o identificador da vulnerabilidade, severidade, descrição, ativo afetado e ferramenta de origem, facilitando a análise posterior e a apresentação consolidada dos resultados. Após isto, existe uma etapa de deduplicação, necessária para lidar com situações em que a mesma vulnerabilidade é identificada por diferentes ferramentas.

Após a normalização e deduplicação, o *SAS* agrega os resultados numa estrutura final orientada ao contexto do ativo analisado. Esta estrutura inclui vulnerabilidades identificadas, ativos sem resultados relevantes, falhas ocorridas durante a pesquisa e metadados adicionais que permitem interpretar a origem da informação recolhida.

A implementação do serviço segue uma lógica *stateless*, evitando persistência própria para os resultados de consolidação. Em vez disso, o *SAS* executa a pesquisa sobre as fontes externas no momento da invocação e produz resultados temporários, que podem ser apresentados diretamente ao utilizador ou disponibilizados para integração com outros componentes do sistema.

Para reforçar robustez durante a execução, esta funcionalidade inclui mecanismos de *retry* e tratamento de falhas transitórias nas integrações externas. Desta forma, o impacto dos *rate-limits* das *APIs* consultadas é reduzido e melhora a estabilidade do processo.

4.4 Integração com a CMDB

A integração com a *CMDB* constitui um elemento central na implementação do *Security Command Hub*, porque o inventário corporativo funciona como fonte de referência para a identificação de ativos, classificação de contexto e suporte ao planeamento de *assessments*.

No *SAS*, a *CMDB* é utilizada como fonte de atributos necessários para a pesquisa de vulnerabilidades. Sempre que o serviço recebe um pedido associado a um ativo ou a um contexto proveniente do *Jira*, a implementação consulta a *CMDB* para obter informação que permite identificar o ativo e determinar quais as ferramentas de segurança relevantes para o caso analisado.

No *RTF*, a integração com a *CMDB* assume um papel ainda mais central, porque o planeamento de *assessments* depende diretamente da informação sincronizada a partir do inventário corporativo. A implementação mantém localmente um conjunto de entidades derivadas da *CMDB*, permitindo que a plataforma disponha de informação consistente sobre ativos, empresas, relações organizacionais e outros dados relevantes para o planeamento.

Esta sincronização é realizada de forma automática através de processos periódicos executados pelo *scheduler* do sistema. Durante este processo, a aplicação obtém informação atualizada dos ativos e das estruturas associadas, atualizando os registos persistidos localmente para garantir que o planeamento se baseia em dados recentes.

A implementação do *RTF* não trata a *CMDB* apenas como fonte de consulta pontual. Em vez disso, a plataforma mantém uma representação persistente do inventário necessário ao seu domínio, permitindo que o cálculo de prioridades, tipos de teste e sugestões de planeamento seja realizado sobre dados locais, já estruturados para o modelo do sistema.

Do ponto de vista de implementação, esta integração foi realizada através de chamadas a *APIs* externas, encapsuladas na lógica de cada serviço. Esta opção permite desacoplar a aplicação da estrutura interna da *CMDB* e concentrar no domínio da solução a lógica de transformação necessária para adaptar a informação do inventário ao modelo utilizado pelos serviços.

4.5 Orquestrador RTF

A implementação do *RTF* não se limita ao armazenamento de informação sobre ativos e *assessments*. O sistema atua também como mecanismo de orquestração dos processos necessários para manter o planeamento atualizado, materializar avaliações no *Jira* e coordenar a execução de determinados tipos de *assessments*.

Esta responsabilidade de orquestração resulta da necessidade de articular diferentes operações sobre dados persistidos, integrações externas e regras de negócio. Na prática, o *RTF* combina operações iniciadas pelos utilizadores com processos automáticos executados em *background*, assegurando que a informação usada no planeamento permanece atualizada e que os *assessments* podem ser convertidos em ações operacionais concretas.

A implementação desta lógica foi concentrada no *backend* do *RTF*, onde coexistem mecanismos de persistência, serviços e tarefas periódicas coordenadas por um *scheduler*. Esta abordagem permite centralizar no mesmo componente a execução dos algoritmos de planeamento, a atualização do estado do sistema e a integração com serviços externos como a *CMDB*, o *Jira* e o *SAS*.

4.5.1 Processos

A implementação do orquestrador do *RTF* inclui um conjunto de processos automáticos responsáveis por manter a informação sincronizada, recalcular dados de suporte ao planeamento e executar tarefas operacionais associadas aos *assessments*. Estes processos são executados através de um *scheduler* integrado no *backend* da aplicação.

A utilização de um *scheduler* permite que o sistema execute tarefas recorrentes sem intervenção do utilizador, assegurando que a plataforma se mantém atualizada e que o planeamento de *assessments* reflete a informação mais recente disponível no sistema.

Um dos processos centrais corresponde à atualização periódica da informação necessária ao funcionamento do *RTF*. Este processo inclui a atualização de ativos, limpeza e renovação de dados transitórios, atualização de vulnerabilidades associadas ao contexto da plataforma e recálculo de métricas utilizadas no planeamento. Esta operação garante que o sistema trabalha sobre dados recentes e coerentes com o estado atual do inventário e das avaliações.

Outro processo relevante corresponde à criação de *assessments* no *Jira*. Quando determinadas condições de planeamento são satisfeitas, o *RTF* pode criar um *assessment* planeado através da criação do respetivo *ticket* no *Jira*. Este processo traduz uma decisão já registada na plataforma numa ação operacional concreta, integrando o planeamento interno com o fluxo de trabalho utilizado pela Euronext.

A implementação inclui também processos associados à execução de *Vulnerability Assessments*. Nestes casos, o *RTF* valida se os *assessments* se encontram num estado compatível com execução automática e, quando aplicável, desencadeia a integração com o *SAS*. Esta operação permite recolher vulnerabilidades associadas ao ativo e registar os resultados no contexto do *assessment* correspondente.

Para além da execução dos processos, a implementação inclui mecanismos de controlo sobre a concorrência das tarefas automáticas. Como a aplicação foi concebida para funcionar em ambiente distribuído, a coordenação do *scheduler* garante que as tarefas periódicas não são executadas de forma duplicada por múltiplas instâncias da aplicação.

A combinação entre processos automáticos e operações iniciadas pelo utilizador permite ao *RTF* suportar um modelo de funcionamento híbrido. O sistema automatiza a atualização da informação, o cálculo de suporte ao planeamento e a execução de determinadas tarefas operacionais, mas mantém sob o controlo humano a criação efetiva de *assessments* e a validação das decisões de planeamento.

4.6 Persistência

A persistência de dados do *Security Command Hub* foi concentrada no *RTF*, refletindo a separação de responsabilidades definida na arquitetura da solução. Enquanto o *SAS* executa pesquisas de vulnerabilidades de forma *stateless*, o *RTF* mantém a informação necessária para suportar o planeamento, histórico de *assessments* e a rastreabilidade operacional necessária.

A implementação no *RTF* utiliza uma base de dados relacional em *MySQL*, acedida através da biblioteca *SQLAlchemy* em modo assíncrono. Esta base de dados encontra-se alojada numa *Amazon Relational Database Service* (*RDS*), integrando-se na arquitetura *cloud* da solução e permitindo separar a camada de persistência da execução dos serviços aplicacionais.

A persistência local do *RTF* inclui entidades relacionadas com ativos, empresas, tipos de *assessment*, prioridades, *assessments*, vulnerabilidades e estruturas auxiliares de suporte ao planeamento. Esta organização permite representar de forma persistente o estado do sistema e manter histórico suficiente para suportar algoritmos, auditoria e análise longitudinal.

A inicialização da persistência é tratada automaticamente pelo *backend* da aplicação. No arranque do sistema, são criadas as estruturas necessárias na base de dados e carrega dados base utilizados pelo domínio, garantindo que o ambiente de execução fica preparado para suportar o funcionamento da plataforma.

A utilização da *RDS* complementa a execução dos serviços aplicacionais em *containers*, assegurando uma separação clara entre camada de aplicação e camada de dados. Esta organização facilita a evolução futura da solução e reforça o alinhamento entre a implementação e a arquitetura física definida para o sistema.

4.7 Segurança e Escalabilidade

A implementação do *Security Command Hub* considera mecanismos de segurança ao nível do acesso às aplicações, proteção das *APIs* e separação entre componentes. Desta forma, procuramos reduzir a exposição indevida da solução e garantir o controlo sobre as operações executadas por utilizadores e serviços.

O acesso às duas aplicações é realizado através de autenticação com *Single Sign-On* (*SSO*), permitindo integrar a solução com o *Identity Provider* já existente da Euronext. No caso do *RTF*, inclui também um controlo de permissões com base em roles, permitindo restringir operações de acordo com o perfil do utilizador autenticado.

As *APIs* das duas aplicações encontram-se protegidas por autenticação, garantindo que apenas pedidos autenticados podem aceder à lógica de negócio.

Do ponto de vista de escalabilidade, a execução das aplicações em *containers* independentes permite ajustar o *frontend* e o *backend* de forma separada, de acordo com a carga observada

em cada serviço. Esta abordagem é particularmente útil no caso do *SAS*, onde o volume de integrações com ferramentas externas pode variar consoante o número de pesquisas executadas.

4.8 Deployment

O *deployment* do *Security Command Hub* foi implementado através de pipelines de integração e publicação definidas nos repositórios *Gitlab* do *SAS* e do *RTF*.

Os dois repositórios utilizam uma estrutura de *pipeline* semelhante, organizada segundo os estágios de *build*, *test*, *publish* e *sign*. A pipeline principal atua como mecanismo de orquestração e desencadeia pipelines específicas para *frontend* e *backend*, de acordo com os ficheiros alterados em cada componente. Esta estrutura está presente tanto no *RTF* como no *SAS*.

Durante a execução, as aplicações são compiladas em imagens *Docker* independentes para *frontend* e *backend*. Estas imagens são publicadas e assinadas, garantindo a integridade dos artefactos gerados e permitindo que apenas versões controladas sejam promovidas para *deployment*.

Após isto, o processo de *continuous delivery* continua num repositório separado, pertencente à equipa de *cloud* da Euronext, responsável por manter os *templates* de *deployment* das aplicações. Neste repositório são atualizadas as referências às assinaturas das imagens produzidas, permitindo associar explicitamente cada *deployment* a um artefacto previamente construído e assinado.

Esta separação entre a construção das imagens e a definição do *deployment* permite desacoplar o processo de *build* do processo de disponibilização da aplicação na infraestrutura. Ao mesmo tempo, introduz um ponto adicional de controlo sobre as versões promovidas para execução, porque o *deployment* apenas avança depois da atualização e integração no repositório que contém os *templates*.

Quando a alteração nesse repositório é submetida através de um *merge request* e posteriormente integrada, o processo de *deployment* é desencadeado automaticamente. Desta forma, mantemos um fluxo controlado de atualizações para ambiente produtivo, reduzindo a intervenção manual e assegurando consistência entre os artefactos produzidos e os *templates* utilizados para *deployment*.

A Figura 28 apresenta uma visão simplificada do fluxo de execução das pipelines de *build*, assinatura e *deployment*.

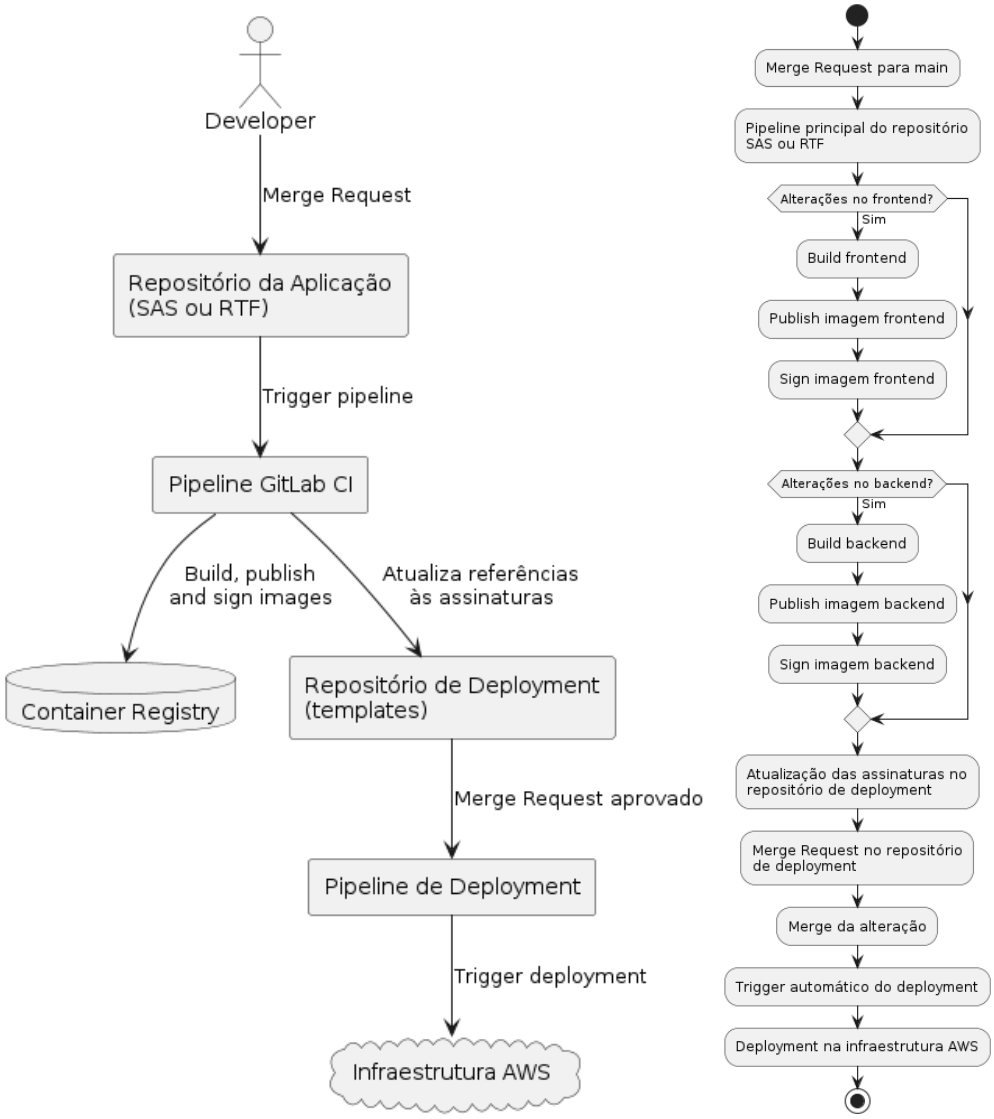


Figura 28 – Fluxo de execução das pipelines de *build*, assinatura e *deployment*

5 Avaliação Experimental

Este capítulo descreve a avaliação experimental realizada sobre o protótipo do *Security Command Hub*. O objetivo desta avaliação consiste em analisar em que medida a solução implementada responde ao problema identificado, utilizando como referência comparativa o processo manual anteriormente utilizado pela equipa de segurança da Euronext.

A avaliação não foi concebida como estudo estatístico em ambiente produtivo. O trabalho incidiu sobre um protótipo funcional, avaliado através de cenários de teste controlados, observação direta do comportamento dos componentes e análise estrutural da arquitetura implementada. Esta opção metodológica permite validar propriedades técnicas e funcionais centrais da solução, mas não permite medir de forma conclusiva o seu impacto quantitativo em produção nem a sua estabilidade ao longo de ciclos operacionais prolongados.

Neste contexto, a avaliação foi organizada em cinco dimensões: eficiência operacional, qualidade dos dados, qualidade do planeamento, rastreabilidade e evidência relevante para o *DORA*, e escalabilidade. Cada dimensão foi analisada segundo o tipo de validação que o protótipo permite sustentar, distinguindo entre comportamento diretamente observável, validação por cenários controlados e análise estrutural da arquitetura.

5.1 Metodologia de Avaliação

A metodologia de avaliação baseia-se na execução de cenários de teste representativos sobre o protótipo funcional do *Security Command Hub*. Cada cenário foi definido para validar propriedades concretas da solução, nomeadamente consolidação de vulnerabilidades, associação a ativos, coerência do planeamento, rastreabilidade operacional e articulação entre os módulos *SAS* e *RTF*.

A avaliação utiliza como *baseline* o processo manual anteriormente descrito, caracterizado por consultas separadas a múltiplas ferramentas, reconciliação manual com a *CMDB*, consolidação

em folhas *Excel* e criação manual de *tickets*. O protótipo é comparado com essa *baseline* ao nível da estrutura do fluxo, da coerência dos resultados e da capacidade de produzir evidência operacional.

As conclusões apresentadas neste capítulo baseiam-se em evidência observável recolhida a partir dos componentes do sistema. Essa evidência inclui respostas produzidas pelo *SAS*, registos persistidos no *RTF*, *tickets* gerados no *Jira*, histórico de *assessments* e comportamento dos fluxos operacionais perante os cenários definidos.

Para evitar confusão entre tipos de evidência, a avaliação distingue explicitamente o âmbito de validação atingível em cada dimensão, visível na Tabela 10.

Tabela 10 – Âmbito de validação por dimensão

Dimensão	Nível de validação atingível	O que não é possível demonstrar com o protótipo
Eficiência operacional	Estrutural e funcional	Medição temporal em produção
Qualidade dos dados	Funcional	Comparação quantitativa com *datasets* históricos reais
Qualidade do planeamento	Funcional	Validação longitudinal ao longo de ciclos reais
Rastreabilidade e evidência relevante para o *DORA*	Funcional	Observação de ciclos reais de auditoria e conformidade
Escalabilidade	Funcional	Testes de carga extensivos e *throughput* em produção

Esta distinção é importante porque o trabalho valida sobretudo dois níveis. O primeiro é a viabilidade técnica da abordagem proposta. O segundo é a correção funcional dos principais mecanismos implementados. Em contrapartida, o trabalho não pretende demonstrar impacto quantitativo completo em produção nem comportamento longitudinal da solução ao longo do tempo.

A Tabela 11 apresenta os cenários de teste utilizados na avaliação.

Tabela 11 – Cenários de teste utilizados na avaliação

Cenário	Descrição	Componentes envolvidos	Objetivo de validação
C1	Pesquisa de vulnerabilidades para um ativo associado a múltiplas fontes	SAS, CMDB	Validar consolidação multi-fonte e associação a ativo
C2	Pesquisa com vulnerabilidades repetidas em mais do que uma fonte	SAS	Validar deduplicação e preservação da origem
C3	Ativo crítico e exposto à internet	RTF	Validar prioridade e tipo de teste

C4	Ativo com histórico recente de avaliação	RTF	Validar uso do histórico no agendamento
C5	Ativo com data de agendamento manual futura válida	RTF	Validar preservação da intervenção manual
C6	Criação de um *Vulnerability Assessment* com execução via *SAS*	RTF, SAS, Jira	Validar integração entre planeamento, execução e *ticketing*
C7	Fluxo completo entre ativo, vulnerabilidade, *assessment* e evidência operacional	SAS, RTF, Jira, CMDB	Validar rastreabilidade *end-to-end*

5.2 Ambiente Experimental

A avaliação foi realizada sobre um protótipo funcional do *Security Command Hub*, composto pelo *SAS* e pelo *RTF*, integrados com a *CMDB*, com o *Jira* e com as ferramentas de segurança utilizadas pela Euronext.

O ambiente experimental procurou reproduzir, de forma controlada, os principais fluxos operacionais da solução, incluindo pesquisa de vulnerabilidades por ativo, consolidação multi-fonte, sincronização de ativos, geração de sugestões de planeamento, criação de *assessments* e execução de *Vulnerability Assessments*.

A evidência recolhida durante a avaliação incluiu quatro tipos principais de observação. O primeiro tipo correspondeu a respostas funcionais produzidas pelo *SAS* durante a pesquisa e consolidação de vulnerabilidades. O segundo tipo correspondeu a registos e entidades persistidas pelo *RTF* durante o planeamento e criação de *assessments*. O terceiro tipo correspondeu a *tickets* e atualizações produzidas pelo *Jira*. O quarto tipo correspondeu ao comportamento dos fluxos operacionais e dos algoritmos perante os cenários definidos.

A Tabela 12 apresenta os tipos de evidência utilizados ao longo da avaliação.

Tabela 12 – Tipos de evidência utilizados ao longo da avaliação

Tipo de evidência	Origem	Utilização na avaliação
Resposta consolidada de vulnerabilidades	SAS	Validar consolidação, normalização, deduplicação e associação ao ativo
Registos persistidos de planeamento	RTF	Validar prioridade, tipo de teste, datas sugeridas e histórico
Tickets e atualizações	Jira	Validar materialização operacional e rastreabilidade
Comportamento dos fluxos e algoritmos	SAS e RTF	Validar coerência da lógica implementada nos cenários definidos

A avaliação incidiu sobre cenários representativos do contexto organizacional, mas não corresponde a utilização contínua em ambiente produtivo. Os resultados devem, por isso, ser interpretados como validação experimental do protótipo e não como medição definitiva do impacto da solução em produção.

A avaliação incluiu também uma componente complementar de observação técnica do comportamento das *APIs* do *SAS* e do *RTF*. Para esse efeito, foram analisadas métricas de execução dos respetivos containers em *AWS ECS*, incluindo utilização de *CPU*, memória, rede e número de *tasks* ativas, bem como realizados testes de carga controlados com recurso ao *Apache JMeter*. Estes testes incidiram sobre as *APIs* de ambos os serviços, com 100 utilizadores concorrentes, valor considerado representativo para o contexto esperado de utilização simultânea da solução.

A Tabela 13 apresenta os testes complementares de escalabilidade e desempenho realizados.

Tabela 13 – Testes complementares de escalabilidade e desempenho

Serviço	Ferramenta	Cenário	Objetivo
SAS	*AWS ECS Container Insights*	Observação de métricas do container	Analisar estabilidade de execução
RTF	*AWS ECS Container Insights*	Observação de métricas do container	Analisar estabilidade de execução
SAS	*Apache JMeter*	100 utilizadores concorrentes	Avaliar resposta da *API* sobre carga controlada
RTF	*Apache JMeter*	100 utilizadores concorrentes	Avaliar resposta da *API* sobre carga controlada

5.3 Avaliação por Dimensão

A avaliação da solução foi organizada segundo cinco dimensões principais, alinhadas com as questões de investigação e com as métricas definidas na estratégia de avaliação do trabalho. Em cada dimensão, o texto distingue o tipo de evidência recolhida e o respetivo alcance.

5.3.1 Eficiência Operacional

A eficiência operacional foi avaliada com o objetivo de verificar se a solução reduz a fragmentação do processo, a dependência de tarefas manuais e o esforço necessário para transformar informação técnica em ações operacionais concretas. Esta dimensão considera não apenas a pesquisa de vulnerabilidades, mas também o apoio ao planeamento de *assessments* e a materialização desse planeamento no *Jira.*

No processo manual, a equipa de segurança consulta a *CMDB*, acede a várias ferramentas de segurança, consolida os resultados em folhas *Excel*, interpreta manualmente o contexto de cada ativo e atualiza os *tickets* e o planeamento em sistemas separados. No fluxo suportado

pelo protótipo, parte significativa destas etapas passa a estar concentrada entre o *SAS*, o *RTF* e o *Jira*.

Nesta dimensão, a avaliação não procurou medir tempo poupado em produção. Procurou antes validar se a estrutura do fluxo suportado pela solução reduz dispersão operacional e substitui etapas manuais intermédias por operações internas do sistema.

A Tabela 14 apresenta os critérios considerados nesta dimensão.

Tabela 14 – Critérios de Eficiência Operacional

Critério	Descrição	Evidência observável	Classificação
Consolidação do fluxo de recolha	O sistema executa a recolha sem exigir consulta manual a múltiplas ferramentas	A pesquisa é iniciada numa única interface e o resultado é devolvido de forma consolidada	Verificado diretamente
Apoio automático ao planeamento	O sistema calcula prioridade, tipo de teste e sugestão de data sem construção manual prévia do plano	O *RTF* apresenta sugestões já calculadas para o ativo	Verificado por cenário controlado
Continuidade entre análise e planeamento	A informação produzida pelo *SAS* pode ser utilizada pelo *RTF* sem transformação manual adicional	O fluxo entre consolidação e planeamento ocorre dentro da solução	Verificado por cenário controlado
Materialização operacional do *assessment*	O sistema converte um *assessment* planeado numa ação concreta no *Jira*	A criação do *assessment* origina o *ticket* correspondente	Verificado diretamente
Redução de fragmentação operacional	O fluxo proposto concentra atividades antes distribuídas por múltiplos artefactos	A arquitetura e os fluxos implementados eliminam reconciliação intermédia em *Excel*	Suportado estruturalmente

A Figura 29 apresenta uma comparação simplificada entre o processo manual e o fluxo suportado pelo Security Command Hub, evidenciando a diferença entre um processo distribuído por múltiplas ferramentas e um fluxo concentrado nas aplicações propostas.

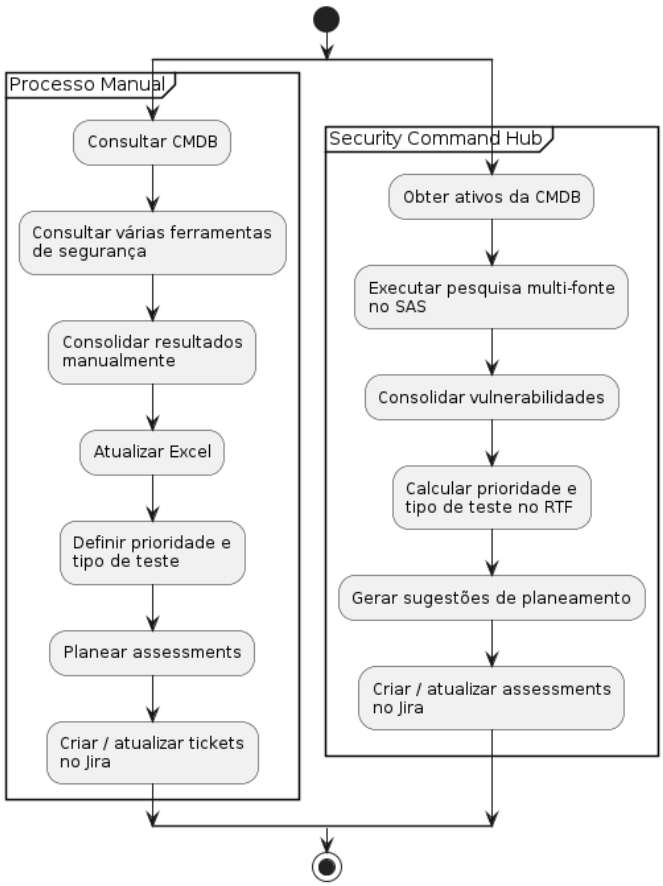


Figura 29 – Comparação entre o processo manual e o fluxo suportado pelo *Security Command Hub*

A avaliação demonstra que o protótipo suporta um fluxo operacional mais estruturado do que o processo manual, sobretudo na passagem entre recolha de vulnerabilidades, planeamento e criação de *assessments*. No entanto, esta secção não mede quantitativamente o impacto temporal dessa mudança em contexto produtivo. Por essa razão, a evidência aqui apresentada deve ser lida como validação funcional e estrutural, e não como medição de eficiência em produção.

5.3.2 Qualidade dos Dados

A qualidade dos dados foi avaliada com o objetivo de verificar se a solução produz uma resposta consolidada, coerente e utilizável a partir de múltiplas fontes heterogéneas. Esta dimensão é central porque o processo atual depende de fontes distintas e de consolidação manual, o que favorece duplicações, omissões e inconsistências.

A avaliação desta dimensão recorreu aos cenários C1 e C2, em que um mesmo ativo pode estar associado a resultados provenientes de múltiplas ferramentas e em que a mesma vulnerabilidade pode surgir repetidamente em fontes diferentes.

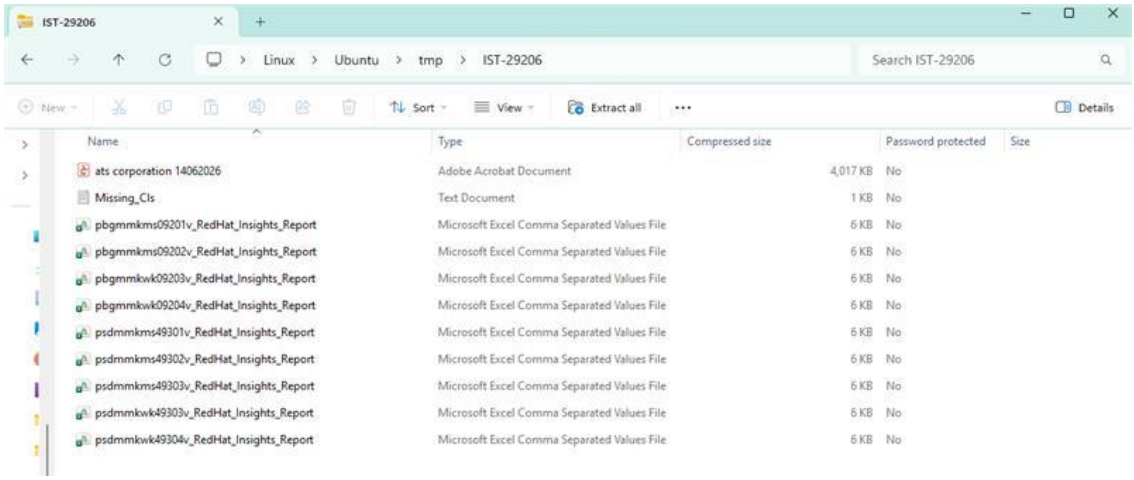


Figura 30 – Exemplo dos resultados do *SAS* face a um *assessment* (IST-29206)

Asset Name	OS	CVE	Impact	Description	CVSS3 Score
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2019-1125	Moderate	A Spectre gadget was found in the Linux kernel's implementation of syst	5.9
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2024-34459	Low	A flaw was found in the xmllint program distributed by the libxml2 packag	5.5
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2024-4741	Low	A use-after-free vulnerability was found in OpenSSL. Calling the OpenSS	5.6
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2024-41073	Moderate	In the Linux kernel, the following vulnerability has been resolved:	5.5
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-21858	Moderate	A use-after-free vulnerability exists in the Linux kernel. When dev_net is	7.3
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-9714	Moderate	A flaw was found in libxstl/libxml2. The 'exsltDynMapFunction' function in	6.2
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-10911	Moderate	A use-after-free vulnerability was found in libxslt while parsing xsl nodes	5.5
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-39981	Moderate	A flaw was found in the Linux kernel's Bluetooth management subsystem	7.3
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-40252	Moderate	QLogic qede driver processes TPA (TCP/IP Packet Aggregation) completi	4.7
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-14087	Moderate	A flaw was found in GLib (Gnome Lib). This vulnerability allows a remote	5.6
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2023-53781	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.3
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-14512	Moderate	A flaw was found in glib. This vulnerability allows a heap buffer overflow	6.5
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-68183	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-68741	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.3
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-68724	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-68366	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-68347	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2025-71116	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2026-22990	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1
pbgmmkwk09203v.prod.exch.int	RHEL 8.10	CVE-2026-22984	Moderate	In the Linux kernel, the following vulnerability has been resolved:	7.1

Figura 31 – Exemplo dos *findings* encontrados para um *CI* específico

A Tabela 15 e a Tabela 16 apresentam os critérios e os cenários utilizados nesta dimensão.

Tabela 15 – Critérios de Qualidade dos Dados

Critério	Descrição	Evidência observável	Resultado
Consolidação multi-fonte	O sistema agrega resultados de mais do que uma ferramenta para o mesmo ativo	A resposta do *SAS* contém *findings* de múltiplas origens	Verificado diretamente
Associação ao ativo	O sistema associa os resultados ao ativo resolvido a partir da *CMDB*	A resposta final mantém a ligação explícita ao ativo analisado	Verificado diretamente
Normalização	O sistema converte resultados heterogéneos para uma estrutura comum	Os campos principais surgem representados de forma uniforme	Verificado por cenário controlado
Deduplicação	O sistema evita apresentar repetidamente a mesma	A resposta final contém registos consolidados face à entrada bruta	Verificado por cenário controlado

	vulnerabilidade no mesmo contexto		

Tabela 16 – Cenários de validação de Qualidade dos Dados

Cenário	Situação observada	Resultado esperado	Resultado
C1	Mesmo ativo com resultados em múltiplas ferramentas	Resposta única consolidada por ativo	Verificado diretamente
C2	Vulnerabilidade repetida em mais do que uma fonte	Deduplicação visível na resposta final	Verificado por cenário controlado

A avaliação demonstra que o protótipo implementa os mecanismos necessários para produzir uma resposta consolidada e coerente no contexto dos cenários executados. Ainda assim, esta secção não compara o desempenho do sistema com *datasets* históricos reais nem quantifica a redução de inconsistências em produção. Assim, a evidência sustenta a correção funcional dos mecanismos, mas não mede o seu impacto estatístico em larga escala.

5.3.3 Qualidade do Planeamento

A qualidade do planeamento foi avaliada com o objetivo de verificar se a solução produz sugestões de *assessments* coerentes com os atributos dos ativos, com o histórico disponível e com as regras definidas nos algoritmos de priorização, tipo de teste e agendamento.

Figura 32 – Exemplo do planeamento agendado para os ativos

A avaliação foi realizada através da execução de cenários C3, C4 e C5, com ativos de diferentes criticidades, contextos operacionais e históricos de avaliação. Em cada cenário, observou-se se os resultados produzidos pelo *RTF* eram coerentes com as regras formais definidas no capítulo 3.6.

A Tabela 17 apresenta os critérios verificáveis utilizados nesta dimensão.

Tabela 17 – Critérios de Qualidade do Planeamento

Critério	Descrição	Evidência observável	Resultado
Coerência da prioridade	A prioridade sugerida reflete criticidade, exposição e contexto regulamentar	O resultado corresponde às regras definidas para os atributos do ativo	Verificado por cenário controlado

Adequação do tipo de teste	O tipo de teste sugerido é compatível com a natureza e contexto do ativo	O tipo atribuído corresponde ao ramo esperado no algoritmo	Verificado por cenário controlado
Utilização do histórico	A data sugerida considera o último teste, a periodicidade e o estado do registo	O cálculo da data varia em função do histórico disponível	Verificado por cenário controlado
Preservação de intervenção manual	O sistema mantém datas manuais válidas sem recálculo indevido	O resultado preserva a data manual quando esta continua válida	Verificado diretamente

A Tabela 18 apresenta cenários representativos utilizados nesta validação.

Tabela 18 – Cenários de validação de Qualidade do Planeamento

Cenário	Setup	Resultado Esperado	Resultado
C3	Ativo crítico, exposto à internet, com criticidade *DORA* elevada	Prioridade alta e tipo de teste mais exigente	Verificado por cenário controlado
C4	Ativo com histórico recente de avaliação	Agendamento ajustado ao histórico e periodicidade	Verificado por cenário controlado
C5	Ativo com data manual futura válida	Preservação da data manual	Verificado diretamente

A avaliação demonstra que o protótipo produz resultados coerentes com as regras definidas para o planeamento. Esta é uma validação de correção funcional dos algoritmos e não uma validação longitudinal do comportamento do plano em ciclos reais de operação. Por essa razão, a evidência demonstra que o planeamento é executável e explicável no protótipo, mas não mede ainda a estabilidade do plano ao longo do tempo.

5.3.4 Rastreabilidade e Evidência Relevante para o DORA

A rastreabilidade foi avaliada com o objetivo de verificar se a solução permite seguir de forma explícita a relação entre ativo, vulnerabilidade, *assessment* e evidência operacional. Esta dimensão é relevante porque o *DORA* reforça a necessidade de demonstrar controlo sobre o planeamento, execução e evidência dos processos de resiliência digital.

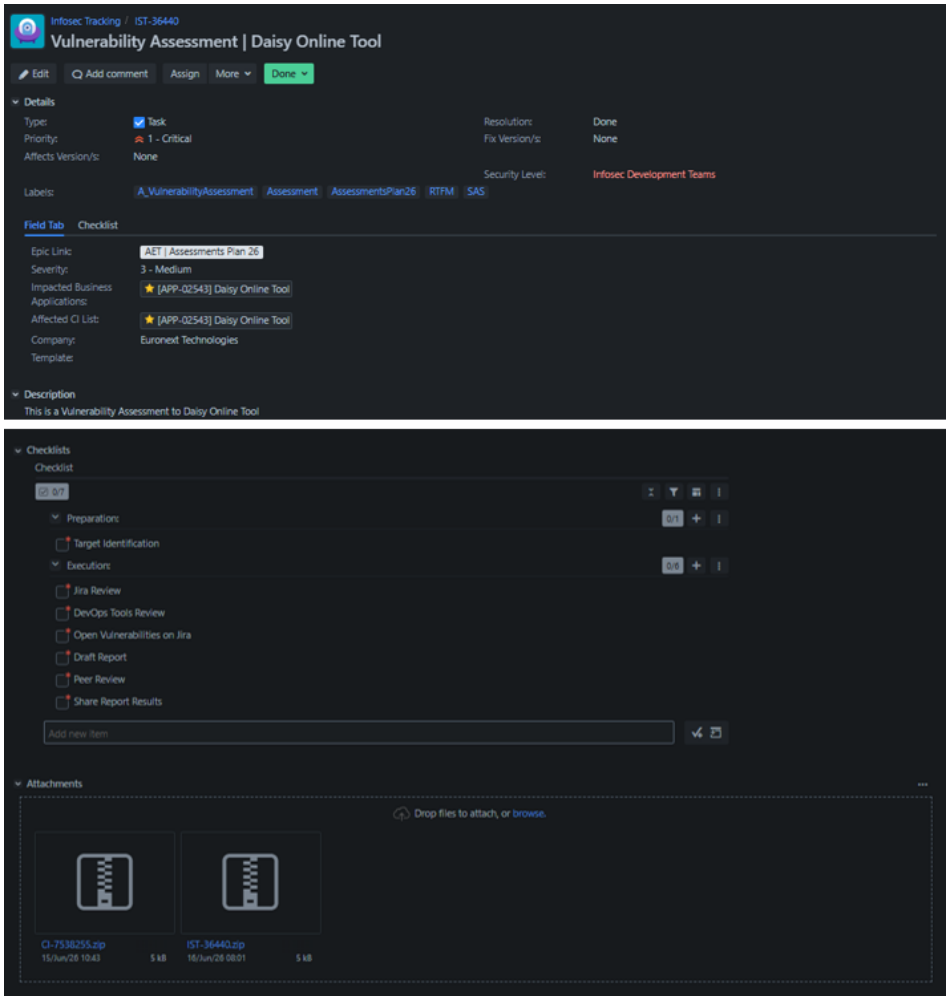


Figura 33 – Exemplo de rastreabilidade e evidência para o DORA

No processo manual, a rastreabilidade depende da manutenção paralela de folhas *Excel*, *tickets* e consultas às ferramentas de segurança. Com a solução proposta, esse fluxo passa a estar articulado entre *CMDB*, *SAS*, *RTF* e *Jira*.

A avaliação incidiu sobretudo sobre os cenários C6 e C7. A Figura 34 apresenta uma visão simplificada do fluxo de rastreabilidade suportado pela solução.

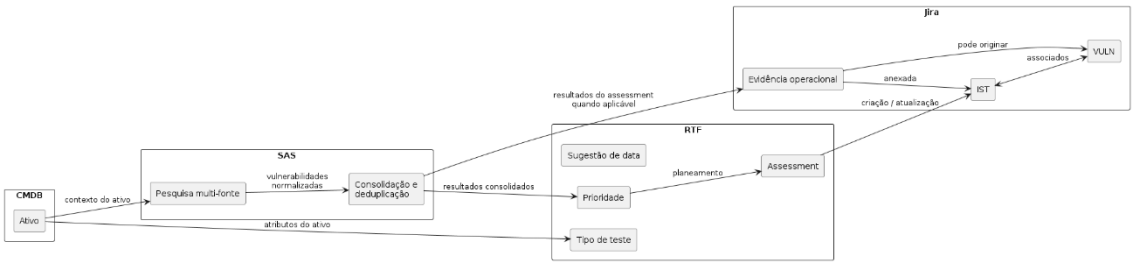


Figura 34 – Fluxo de rastreabilidade entre ativo, vulnerabilidades, *assessment* e evidência operacional

A Tabela 19 apresenta os critérios desta dimensão.

Tabela 19 – Critérios de Rastreabilidade e Evidência Relevante para o DORA

Critério	Descrição	Evidência observável	Resultado
Ligação ativo-vulnerabilidade	A vulnerabilidade consolidada permanece associada ao ativo identificado na *CMDB*	A resposta do *SAS* mantém o ativo como contexto explícito da análise	Verificado diretamente
Ligação vulnerabilidade-*assessment*	O ciclo de avaliação relaciona vulnerabilidades com *assessments* executados ou planeados	O *RTF* mantém associação entre ativo, *assessment* e evidência recolhida	Verificado por cenário controlado
Evidência operacional no *Jira*	O sistema converte resultados técnicos em contexto operacional rastreável	O *ticket* criado ou atualizado no *Jira* preserva a informação relevante	Verificado diretamente
Visibilidade do histórico	O sistema permite observar que ativos foram avaliados e quais permanecem planeados	O *RTF* mantém histórico e estado dos *assessments*	Verificado diretamente
Suporte à auditoria regulatória	A solução cria condições para demonstrar rastreabilidade útil em auditoria	O fluxo *end-to-end* mantém ligação entre inventário, análise e execução	Suportado estruturalmente

A avaliação demonstra que o protótipo suporta rastreabilidade *end-to-end* entre inventário, análise técnica, planeamento e *ticketing* operacional. No entanto, esta secção não demonstra ciclos reais de auditoria nem mede diretamente a redução do esforço de preparação de evidência para produção. Assim, a evidência confirma a existência dos mecanismos necessários para suportar rastreabilidade relevante para o *DORA*, mas não prova conformidade operacional completa em contexto real.

5.3.5 Escalabilidade

A avaliação desta dimensão combinou dois níveis de evidência. O primeiro nível corresponde à análise estrutural da arquitetura implementada, incluindo separação entre *SAS* e *RTF*, desacoplamento entre aplicação e persistência, processamento por unidade de contexto e *pipelines* independentes de *build* e *deployment*. O segundo nível corresponde a evidência experimental limitada, obtida através da observação de métricas de execução dos containers em *AWS ECS* e de testes de carga controlados sobre as *APIs* do *SAS* e do *RTF* com recurso ao *Apache JMeter*.

As métricas observadas no *ECS* indicam comportamento estável dos serviços ao longo do período analisado, com utilização reduzida de *CPU* e memória e manutenção de uma *task* ativa por serviço. Esta observação não demonstra escalabilidade em sentido amplo, mas sugere que

os serviços operam de forma estável na configuração de execução utilizada durante a avaliação, como é possível ver na Figura 35 e Figura 36.

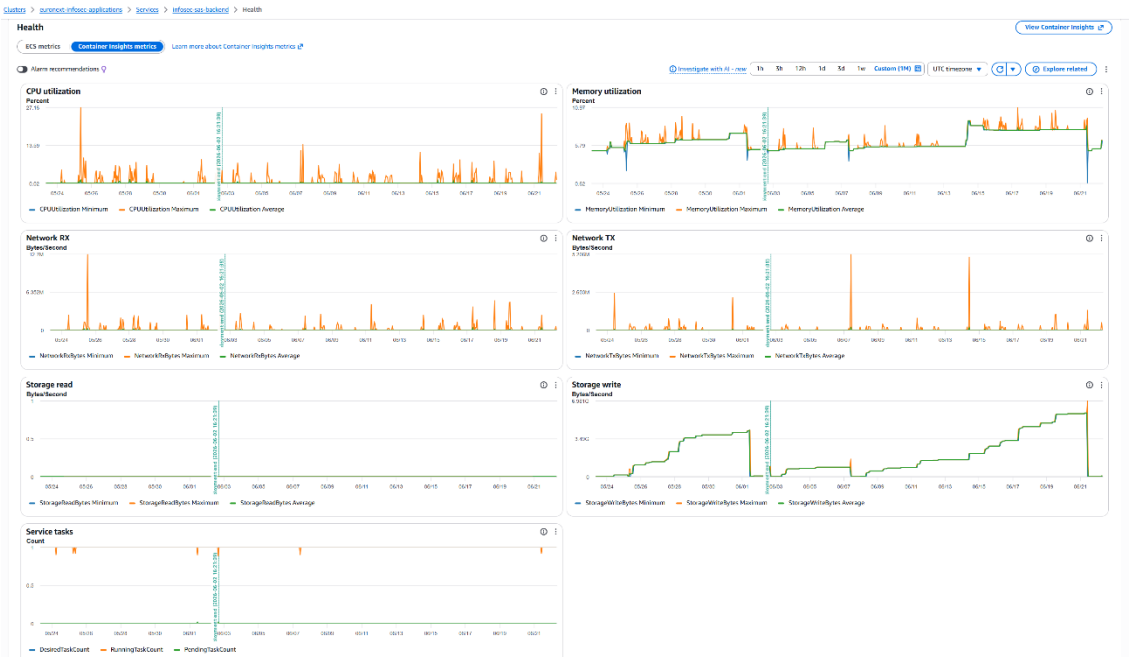


Figura 35 – Métricas de execução do container da *API* do *SAS*

Figura 36 – Métricas de execução do container da *API* do *RTF*

Os testes de carga controlados foram realizados localmente com 100 utilizadores concorrentes, valor considerado compatível com o volume simultâneo esperado de utilização da solução. No caso do *SAS*, o teste registou 2000 amostras, tempo médio de resposta de 3 *ms*, *throughput* de 200,4 pedidos por segundo e taxa de erro de 0%. No caso do *RTF*, o teste registou 2000 amostras,

tempo médio de resposta de 4 *ms*, *throughput* de 196,3 pedidos por segundo e taxa de erro de 0%. Estes resultados (Figura 37 e Figura 38) mostram que, no cenário testado, ambos os serviços responderam de forma estável e com baixa latência.

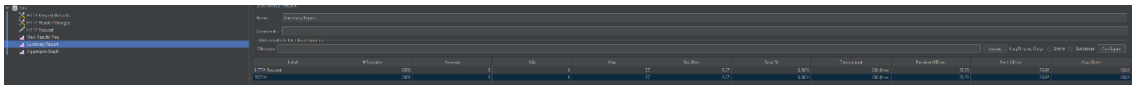


Figura 37 – Resultados dos testes de carga à API do SAS

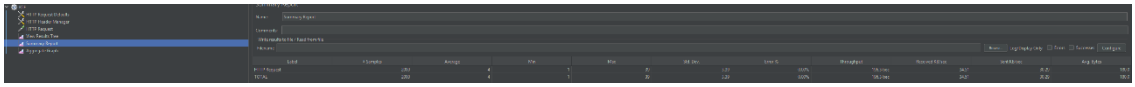


Figura 38 – Resultados dos testes de carga à API do RTF

A Tabela 20 apresenta os critérios utilizados para esta dimensão e o respetivo tipo de evidência.

Tabela 20 – Critérios de Escalabilidade

Critério	Descrição	Evidência observável	Resultado
Separação por serviço	Os componentes podem evoluir e executar de forma independente	*SAS* e *RTF* possuem *frontend* e *backend* separados	Suportado estruturalmente
Desacoplamento entre aplicação e persistência	O crescimento da camada aplicacional não depende de armazenamento local	O *RTF* utiliza uma *RDS* e o *SAS* opera de forma *stateless*	Suportado estruturalmente
Processamento por unidade de contexto	A lógica principal opera por ativo ou por contexto resolvido	Fluxos e algoritmos são aplicados sobre unidades independentes	Suportado estruturalmente
Build e *deployment* desacoplados	Os componentes podem ser promovidos de forma independente	*Pipelines* e fluxo de *deployment* tratam artefactos separadamente	Suportado estruturalmente
Estabilidade básica em execução	Os serviços mantêm comportamento estável na infraestrutura observada	Métricas *ECS* mostram consumo baixo e *task* ativa estável	Verificado diretamente
Resposta da API sob carga controlada	As *APIs* respondem sem erro em cenário concorrente limitado	Testes *JMeter* com 100 utilizadores, 0% de erro em ambos os serviços	Verificado diretamente
Desempenho sob carga elevada e prolongada	O sistema mantém comportamento em produção sob carga contínua	Não foram realizados testes de carga extensivos em produção	Não mensurável no protótipo

A avaliação indica que a solução apresenta condições arquiteturais favoráveis à escalabilidade e evidencia experimental limitada de desempenho satisfatório em cenário controlado. No entanto, esta conclusão deve ser interpretada com cautela. Os testes realizados não equivalem a testes de carga extensivos em produção, nem cobrem de forma exaustiva a variabilidade introduzida por integrações externas, crescimento do volume de ativos ou utilização prolongada do sistema.

5.4 Síntese dos Resultados

A avaliação experimental realizada permitiu observar o comportamento da solução em cinco dimensões complementares: eficiência operacional, qualidade dos dados, qualidade do planeamento, rastreabilidade e evidência relevante para o *DORA*, e escalabilidade. A leitura conjunta dos resultados mostra que o protótipo valida de forma convincente a viabilidade técnica da abordagem proposta e a correção funcional dos seus mecanismos centrais.

As dimensões suportadas por evidência mais forte são a qualidade dos dados, a qualidade do planeamento e a rastreabilidade, porque dependem diretamente de *outputs* observáveis do *SAS*, do *RTF* e do *Jira*. Nestes casos, o protótipo permitiu verificar consolidação multi-fonte, associação entre vulnerabilidades e ativos, coerência dos algoritmos de planeamento e ligação explícita entre inventário, *assessment* e evidência operacional.

A eficiência operacional apresenta evidência intermédia. A avaliação mostrou que a solução substitui um fluxo fragmentado por um processo mais contínuo entre análise, planeamento e execução. No entanto, esta conclusão deve ser entendida como validação funcional e estrutural do fluxo implementado, e não como medição quantitativa do ganho temporal em produção.

A dimensão de escalabilidade e desempenho, que anteriormente assentava sobretudo em análise arquitetural, passou também a incluir evidência experimental limitada. A observação das métricas dos containers do *SAS* e do *RTF* em *AWS ECS* mostrou comportamento estável dos serviços na infraestrutura de execução utilizada, com utilização reduzida de *CPU* e memória e manutenção de uma *task* ativa por serviço. Complementarmente, os testes de carga controlados com recurso ao *Apache JMeter*, realizados com 100 utilizadores concorrentes, mostraram resposta estável das *APIs* dos dois serviços, sem erros e com baixa latência média.

Mais uma vez, estes resultados não equivalem a uma validação completa em produção, nem demonstram comportamento sob carga prolongada ou em condições reais de utilização intensiva. Ainda assim, reforçam a avaliação da solução ao mostrarem que a arquitetura proposta não apenas suporta o desacoplamento e a evolução modular, mas também apresenta estabilidade básica e capacidade de resposta adequada no cenário de carga controlada considerado.

A Tabela 21 apresenta uma síntese global dos resultados observados em cada dimensão de análise.

Tabela 21 – Síntese global dos resultados observados em cada dimensão

Dimensão	Resultado global	Tipo de evidência	Grau de validação
Eficiência operacional	Fluxo mais estruturado e menos fragmentado	Evidência funcional e estrutural	Moderado
Qualidade dos dados	Consolidação, normalização e deduplicação demonstradas	Evidência funcional	Forte
Qualidade do planeamento	Coerência dos algoritmos demonstrada em cenários controlados	Evidência funcional	Forte
Rastreabilidade e evidência relevante para o *DORA*	Fluxo *end-to-end* e ligação entre entidades demonstrados	Evidência funcional e estrutural	Forte
Escalabilidade e desempenho	Condições arquiteturais favoráveis e estabilidade observada sob carga controlada	Evidência estrutural e experimental limitada	Moderado

Em síntese, a avaliação demonstra que o Security Command Hub responde de forma sólida às propriedades que podem ser validadas ao nível de um protótipo funcional. O trabalho demonstra que a abordagem proposta é tecnicamente viável, coerente com o problema identificado e capaz de suportar, de forma observável, mecanismos de integração, consolidação, planeamento e rastreabilidade. Em contrapartida, as dimensões que exigem medição longitudinal em produção ou observação prolongada do uso organizacional permanecem fora do alcance empírico desta dissertação.

5.5 Análise Crítica

A análise crítica dos resultados procura responder às questões de investigação definidas no subcapítulo 1.3.1, relacionando as evidências recolhidas durante a avaliação com o problema que motivou o desenvolvimento do *Security Command Hub*. O objetivo deste subcapítulo não consiste em afirmar que todas as questões ficaram resolvidas de forma plena, mas em analisar de forma realista em que medida a avaliação conduzida permite sustentar respostas a cada uma delas.

A Tabela 22 sintetiza o grau em que a avaliação realizada permite responder às questões de investigação.

Tabela 22 – Grau de resposta relativo à comparação com a avaliação realizada e as questões de investigação

Questão de investigação	Grau de resposta	Fundamentação
QI1 – Automação, rastreabilidade e conformidade	Parcialmente respondida	A avaliação validou ligação entre ativo, vulnerabilidade, *assessment* e *ticket*, mas não mediu ciclos reais de auditoria em produção
QI2 – Consolidação multi-fonte	Respondida	Os cenários de teste validaram consolidação, associação ao ativo e resposta unificada produzida pelo *SAS*
QI3 – Normalização e deduplicação	Parcialmente respondida	Os mecanismos foram validados no protótipo, mas a redução de variabilidade não foi medida quantitativamente em *datasets* históricos
QI4 – Planeamento automático de *assessments*	Parcialmente respondida	Os algoritmos produziram resultados coerentes em cenários controlados, mas não houve validação longitudinal em vários ciclos reais
QI5 – Sincronização bidirecional com a *CMDB*	Parcialmente respondida	A integração e sincronização foram observadas, mas não foi medida quantitativamente a redução de inconsistências ao longo do tempo
QI6 – Evidência operacional e *DORA*	Parcialmente respondida	A centralização de evidência e o fluxo *end-to-end* foram validados, mas a redução efetiva do esforço em contexto produtivo não foi medida

Relativamente à QI1, que procura perceber em que medida a automação do planeamento de testes de segurança e da deteção de vulnerabilidades integrada com a *CMDB* melhora a rastreabilidade operacional e a capacidade de demonstração de conformidade regulatória, os resultados permitem formular uma resposta **positiva, mas parcial**. A avaliação demonstrou que o protótipo suporta uma ligação explícita entre ativo, vulnerabilidade, *assessment* e *ticket* operacional, o que constitui uma melhoria clara face ao processo manual. No entanto, esta resposta permanece incompleta porque a avaliação não acompanhou a utilização da solução em ciclos reais de auditoria, nem mediu diretamente a sua contribuição em processos formais de conformidade.

Relativamente à QI2, centrada na consolidação de dados de vulnerabilidades provenientes de múltiplas fontes heterogéneas, a evidência recolhida permite uma resposta **positiva e suficientemente sustentada ao nível do protótipo**. Os cenários executados mostraram que o

SAS consolida resultados de mais do que uma fonte, preserva a associação ao ativo e devolve uma resposta unificada preparada para consumo no restante sistema. Esta é a questão com evidência mais direta e mais convincente no âmbito do protótipo, porque depende de comportamento funcional observável e verificável.

Relativamente à QI3, que procura avaliar em que medida a normalização e a deduplicação reduzem variabilidade e inconsistência na avaliação do risco, a resposta deve ser considerada **positiva, mas limitada**. A avaliação confirmou que o sistema aplica uma estrutura comum aos resultados recolhidos e elimina duplicações no contexto da resposta consolidada. Contudo, a redução efetiva da variabilidade na avaliação do risco não foi comparada com *datasets* históricos nem com avaliações humanas independentes ao longo do tempo. A evidência demonstra, assim, a existência dos mecanismos necessários para suportar maior consistência, mas não permite quantificar plenamente o seu impacto.

Relativamente à QI4, sobre a automação do cálculo de periodicidade e do agendamento de testes com base na criticidade dos ativos, a resposta é **positiva ao nível da coerência lógica do planeamento**. Os cenários executados mostraram que os algoritmos de prioridade, tipo de teste e agendamento produzem resultados compatíveis com as regras definidas e adaptados ao contexto do ativo, incluindo criticidade, exposição, histórico e decisões manuais. Ainda assim, a resposta continua parcial, porque não foi realizada uma análise longitudinal suficientemente extensa para medir desvio entre plano e execução ao longo de múltiplos ciclos reais de operação.

Relativamente à QI5, centrada na sincronização bidirecional com a *CMDB* e na redução de inconsistências entre inventário, vulnerabilidades e processos de mitigação, a resposta deve ser considerada **parcialmente positiva**. A solução demonstrou viabilidade de integração com a CMDB e utilização efetiva do inventário como base para planeamento e articulação com vulnerabilidades. No entanto, a avaliação não mediu quantitativamente a redução de inconsistências antes e depois da introdução, nem observou o comportamento do sistema ao longo de um período prolongado de sincronização real.

Relativamente à QI6, que procura perceber em que medida a agregação centralizada de informação e evidência operacional reduz o esforço de preparação de evidência e melhora a rastreabilidade exigida pelo *DORA*, a resposta é também **positiva, mas parcial**. A avaliação demonstrou que a solução centraliza informação que antes se encontrava dispersa entre *CMDB*, *tools*, folhas *Excel* e *tickets*, e que o fluxo *end-to-end* permite produzir evidência operacional mais estruturada. Contudo, a redução efetiva do esforço de preparação de evidência não foi medida com utilizadores em contexto produtivo, pelo que a conclusão permanece necessariamente limitada.

A dimensão de desempenho e escalabilidade merece também uma leitura crítica própria. Nesta dissertação, esta dimensão deixou de assentar apenas em análise estrutural da arquitetura, passando a incluir evidência experimental limitada. A observação das métricas dos *containers* em *AWS ECS* e os testes de carga controlados com 100 utilizadores concorrentes forneceram

evidência adicional sobre estabilidade básica e capacidade de resposta das *APIs* do *SAS* e *RTF*. Ainda assim, esta evidência não substitui testes de carga prolongados em produção, nem permite concluir sobre comportamento sob utilização intensiva, crescimento continuado do volume de ativos ou variações induzidas por integrações externas.

De forma global, a avaliação realizada demonstra que o protótipo valida de forma credível a viabilidade técnica e a correção funcional da abordagem proposta. As questões mais diretamente ligadas ao comportamento observável do sistema receberam evidência forte. As questões que implicam medição longitudinal, impacto quantitativo em produção ou observação prolongada em contexto real ficaram apenas parcialmente respondidas.

Esta conclusão não enfraquece o trabalho. Delimita o seu alcance real. A dissertação demonstra que a abordagem proposta é implementável, coerente com o problema e tecnicamente capaz de suportar os mecanismos necessários para integração, consolidação, planeamento e rastreabilidade. Uma validação completa do impacto operacional da solução exigiria, contudo, utilização prolongada em ambiente produtivo, recolha sistemática de métricas ao longo do tempo e observação direta de ciclos reais de operação e auditoria.

6 Conclusões

O presente capítulo encerra o trabalho realizado, sintetizando os principais resultados obtidos e discutindo o alcance real da solução proposta no contexto do problema identificado. A análise final procura consolidar a relação entre os objetivos, a implementação e a avaliação, distinguindo de forma clara o que foi validado no protótipo e o que permanece dependente de evolução futura.

Numa primeira fase, são discutidas as limitações da solução e da avaliação realizada, com o objetivo de enquadrar corretamente o alcance das conclusões apresentadas. De seguida, é analisado o impacto organizacional potencial do *Security Command Hub* no contexto da Euronext. Por fim, são identificadas ameaças à validade da solução e apresentadas direções de trabalho futuro que poderão aprofundar, maturar e melhorar a solução proposta.

6.1 Limitações

A solução proposta apresenta limitações que importa reconhecer. A primeira limitação resulta da dependência da qualidade do inventário mantido na *CMDB*. Quando os dados dos ativos se encontram incompletos, desatualizados ou inconsistentes, a qualidade da consolidação de vulnerabilidades e do planeamento de *assessments* degrada-se, porque ambos os módulos utilizam esse inventário como base de contexto e fonte de verdade.

A segunda limitação resulta da dependência de integrações externas. O *SAS* e o *RTF* comunicam com múltiplas *APIs*, incluindo *CMDB*, *Jira* e ferramentas de segurança. O comportamento da solução permanece, assim, condicionado pela disponibilidade dessas plataformas, pela latência das respostas, por limitações de quota, por requisitos de autenticação e por alterações futuras nos contratos das *APIs*.

A principal limitação metodológica do trabalho resulta do facto de a avaliação ter incidido sobre um protótipo funcional em ambiente controlado. Neste contexto, a dissertação demonstrou

viabilidade técnica, correção funcional dos mecanismos centrais e evidência experimental limitada de desempenho sob carga controlada, mas não demonstrou impacto operacional completo em produção. Em particular, não foi possível observar o comportamento da solução ao longo de ciclos prolongados de utilização, medir quantitativamente a redução de esforço manual em contexto real ou validar longitudinalmente a estabilidade dos planos gerados.

Por fim, a arquitetura adotada apresenta outra das limitações, embora favorável à modularidade, integração e evolução independente dos componentes, introduz um custo operacional adicional associado a containers, pipelines, persistência em *cloud* e manutenção contínua das integrações. Esta limitação não compromete a validade da solução, mas deve ser considerada na sua adoção e evolução futura.

6.2 Impacto Organizacional

O impacto organizacional do *Security Command Hub* resulta sobretudo da capacidade de substituir um conjunto de processos fragmentados e manuais por um fluxo mais coerente entre inventário, vulnerabilidades, planeamento e execução operacional. Esta transformação não elimina o papel da equipa de segurança, mas altera a forma como estas interagem com a informação e com o ciclo de *assessments*.

Ao nível operacional, a solução apresenta potencial, validado em protótipo, para reduzir a dependência de folhas *Excel*, consultas dispersas a múltiplas ferramentas e reconciliação manual com a *CMDB*. Esta mudança tende a melhorar a continuidade do processo, reduzir esforço repetitivo e aumentar previsibilidade no planeamento de avaliações de segurança.

Ao nível da governação, a solução reforça a ligação entre ativos, vulnerabilidades, *assessments* e *tickets* operacionais. Esta propriedade é particularmente relevante num contexto regulado, porque melhora a capacidade de produzir evidência rastreável e de demonstrar controlo sobre o processo de avaliação de segurança.

Ao nível tecnológico, a solução preserva as ferramentas já utilizadas pela organização e introduz uma camada de integração e automação sobre o ecossistema existente. Esta decisão reduz o impacto de adoção, evita a substituição de plataformas críticas e favorece a evolução progressiva da abordagem proposta.

Para além dos aspetos técnicos observados no protótipo, o desenvolvimento da solução foi acompanhado regularmente por elementos da equipa de segurança da Euronext, o que permitiu recolher um feedback constante sobre o potencial valor organizacional da solução. Esse feedback apontou, sobretudo, para três dimensões consideradas relevantes pela equipa:

- Maior centralização da informação;
- Melhor articulação entre inventário e vulnerabilidades;

- Maior estrutura no processo de planeamento de *assessments*.

Foi também identificada com o protótipo, uma perceção inicial de valor na redução de esforço manual associado à recolha e reconciliação de informação, bem como na possibilidade de ligar de forma mais clara a análise técnica ao respetivo contexto operacional.

Ao mesmo tempo, o feedback inicial sugere que o valor organizacional da solução dependerá da sua integração efetiva nos processos já utilizados pela equipa, da confiança na qualidade dos dados sincronizados a partir da *CMDB* e da estabilidade das integrações com as ferramentas externas. Isto significa que o impacto potencial da solução não depende apenas da sua implementação técnica, mas também da forma como vier a ser adotada e incorporada na operação diária.

6.3 Ameaças à Validade

A primeira ameaça à validade deste trabalho resulta do contexto em que a avaliação foi conduzida. A validação incidiu sobre um protótipo funcional e sobre cenários controlados, o que limita a capacidade de generalizar os resultados para utilização contínua em ambiente produtivo. Esta limitação afeta sobretudo conclusões relacionadas com o desempenho prolongado, esforço operacional em produção e comportamento sob carga elevada.

Outra ameaça à validade resulta da dependência de dados externos. A solução utiliza informação proveniente da *CMDB*, do *Jira* e de múltiplas ferramentas de segurança. Quando estes dados apresentam inconsistências, desatualização ou lacunas, as conclusões obtidas sobre consolidação, planeamento e rastreabilidade podem ser influenciadas por fatores externos ao próprio sistema.

A terceira ameaça à validade resulta da dependência de credenciais e *tokens* de autenticação associados às *APIs* das ferramentas integradas. A validade temporal destes *tokens*, a sua renovação e a disponibilidade efetiva do acesso às *APIs* influenciam diretamente a execução dos fluxos da solução. Em contexto experimental, qualquer limitação neste caso pode afetar a completude dos cenários executados e introduzir variabilidade que não decorre da lógica implementada.

A quarta ameaça à validade decorre da natureza representativa, mas limitada, dos cenários utilizados na avaliação. Os cenários escolhidos foram suficientes para validar propriedades centrais da solução, mas não cobrem toda a variabilidade possível de ativos, vulnerabilidades, combinações de fontes e situações operacionais existentes num contexto organizacional real e completo.

Estas ameaças não invalidam o trabalho realizado, mas delimitam o alcance das conclusões apresentadas. Os resultados devem ser interpretados como evidência de viabilidade técnica e

funcional da abordagem proposta, e não como prova definitiva do seu impacto total em contexto produtivo.

6.4 Trabalhos Futuros

O trabalho desenvolvido abre espaço para evolução futura em várias dimensões da solução, tanto ao nível funcional como ao nível do apoio operacional às equipas de segurança. Estas evoluções não alteram a lógica central do *Security Command Hub*, mas alargam o seu âmbito e reforçam o seu valor no contexto diário da organização.

No caso do *SAS*, uma evolução prevista consiste na integração com mecanismos de inteligência artificial para obtenção de informação sobre *End of Life* e *End of Security Patch* de *softwares* e *hardwares* registados na *CMDB*. Esta capacidade permitirá complementar a análise de vulnerabilidades com informação adicional sobre obsolescência tecnológica e fim de suporte, reforçando a contextualização do risco associado a ativos que permanecem em operação sem cobertura adequada de manutenção ou segurança.

Esta evolução é particularmente relevante porque permitirá utilizar o inventário já existente da *CMDB* como base para uma análise mais ampla do ciclo de vida tecnológico dos ativos. Ao introduzir mecanismos atuais de apoio baseados em inteligência artificial, a solução poderá acompanhar tendências de mercado e reforçar a capacidade da equipa para identificar riscos que não resultam apenas de vulnerabilidades conhecidas, mas também de descontinuação de suporte por parte dos fabricantes.

No caso do *RTF*, uma linha de evolução importante consiste na construção de métricas e *dashboards* orientados à análise dos *assessments* realizados. Esta funcionalidade permitirá observar, de forma mais imediata, indicadores como número de *assessments* por mês, distribuição por estado, distribuição por companhia e distribuição por tipo de *assessment*. Esta visibilidade permitirá melhorar o acompanhamento operacional e apoiar decisões de gestão sobre carga, cobertura e execução do plano.

Outra evolução prevista consiste na análise detalhada de ativos considerados críticos, permitindo validar de forma mais direta os *assessments* e vulnerabilidades associados a esses ativos. Esta funcionalidade poderá reforçar o controlo sobre sistemas mais sensíveis do inventário e melhorar a capacidade de análise focalizada sobre elementos com maior impacto organizacional ou regulamentar.

Está igualmente previsto o desenvolvimento de um mecanismo de geração automática de relatórios em formato *PDF* após a realização de um *Vulnerability Assessment*. Esta capacidade permitirá produzir documentação estruturada com maior detalhe técnico e operacional, reduzindo esforço manual de preparação de evidência e reforçando suporte à rastreabilidade e à auditoria.

Em conjunto, estas evoluções futuras reforçam a ambição da solução como plataforma de apoio contínuo às equipas de segurança, permitindo não apenas automatizar tarefas existentes, mas também introduzir novas capacidades de análise, controlo e produção de evidência. A concretização destas linhas de trabalho contribuirá para alargar o âmbito do *Security Command Hub* e para reforçar o seu papel na gestão de milhares de ativos mantidos na *CMDB*.

Referências

- Agutter, C. (2020). ITIL Foundation Essentials ITIL 4 Edition-The Ultimate Revision Guide. IT Governance Publishing Ltd.

- Al Faruq, B., Herlianto, H. R., Simbolon, S. H., Utama, D. N., & Wibowo, A. (2020). *Integration of ITIL V3, ISO 20000 & iso 27001: 2013forit services and security management system. International Journal, 9(3).* https://doi.org/10.30534/ijatcse/2020/157932020

- Alén, C. (2020). *Development of a CMDB Model to Represent SaaS Organization Assets Relevant for Service Delivery.* https://urn.fi/URN:NBN:fi:aalto-202008235043

- Atlassian (2024). *CMDB documentation. Atlassian Corporation. Retrieved from* https://www.atlassian.com/itsm/it-asset-management/cmdb

- Atlassian (2024). *Jira Service Management documentation. Atlassian Corporation. Retrieved from* https://www.atlassian.com/software/jira/service-management/features

- Axelos (2020). ITIL Foundation: ITIL 4 edition. The Stationery Office.

- Bass, L., Lu, Q., Weber, I., & Zhu, L. (2025). Engineering AI systems: architecture and DevOps essentials. Addison-Wesley Professional.

- Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley.

- Brinkman, B., Flick, C., Gotterbarn, D., Miller, K., Vazansky, K., & Wolf, M. J. (2017). Listening to professional voices: draft 2 of the ACM code of ethics and professional

conduct. Communications of the ACM, 60(5), 105-111. https://doi.org/10.1145/3072528

- Chatterjee, R. (2021). *Red Hat Smart Management and Red Hat Insights. In: Red Hat and IT Security. Apress, Berkeley, CA.* https://doi.org/10.1007/978-1-4842-6434-8_5

- Diogenes, Y., & Janetscheck, T. (2022). Microsoft Defender for Cloud. Microsoft Press.

- ENISA. (2023). *ENISA Threat Landscape 2023. European Union Agency for Cybersecurity. Retrieved from* https://www.enisa.europa.eu/publications/interoperable-eu-risk-management-toolbox

- European Commission (2024). *Digital Operational Resilience Act (DORA) – Implementation guidelines. Retrieved from* https://finance.ec.europa.eu/digital-finance/cyber-resilience_en

- European Union (2022). Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) No 1060/2009, (EU) No 648/2012, (EU) No 600/2014, (EU) No 909/2014 and (EU) 2016/1011 (Text with EEA relevance). Retrieved from https://eur-lex.europa.eu/eli/reg/2022/2554

- Huijbregts, P., Anich, J., & Graves, J. (2023). *Microsoft Defender for Endpoint in Depth.*

- ISO/IEC 27001:2022 (2022). Information security, cybersecurity and privacy protection. International Organization for Standardization. Retrieved from https://www.iso.org/standard/27001

- Kim, G., Debois, P., Willis, J., & Humble, J. (2021). The DevOps Handbook: How to create world-class agility, reliability, and security. IT Revolution.

- Kovacevic, B., & DiCola, N. (2023). Security Orchestration, Automation, and Response for Security Analysts: Learn the secrets of SOAR to improve MTTA and MTTR and strengthen your organization's security posture. Packt Publishing Ltd.

- Manne, T. A. K. (2024). Leveraging AWS Security Hub and GuardDuty for Continuous Threat Intelligence. The Journal of Scientific and Engineering Research, 11, 268-275. Research Link

- Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). *Challenges and solutions when adopting DevSecOps: A systematic review. Information and software technology, 141, 106700.* https://doi.org/10.1016/j.infsof.2021.106700

- Scott, H. S. (2021). *The EU's Digital Operational Resilience Act: Cloud Services & Financial Companies. Available at SSRN 3904113.* https://dx.doi.org/10.2139/ssrn.3904113

- Sohval, B. (2020). A Deep Dive in Scoring Methodology. SecurityScorecard Inc.: New York, NY, USA.

- Souppaya, M., & Scarfone, K. (2022). *Guide to enterprise patch management planning: Preventive maintenance for technology.* https://doi.org/10.6028/NIST.SP.800-40r4

- Walkowski, M., Krakowiak, M., Jaroszewski, M., Oko, J., & Sujecki, S. (2021, September). Automatic CVSS-based vulnerability prioritization and response with context information. In 2021 international conference on software, telecommunications and computer networks (softCOM) (pp. 1-6). IEEE. https://doi.org/10.23919/SoftCOM52868.2021.9559094

- Zaeem, R. N., & Barber, K. S. (2020). The effect of the GDPR on privacy policies: Recent progress and future promise. ACM Transactions on Management Information Systems (TMIS), 12(1), 1-20. https://doi.org/10.1145/3389685