UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Martínez Llobregat, Sara. *Pipeline de dades per priorització de vulnerabilitats (CVE, EPSS, CISA KEV)*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Enginyeria de Dades)

This version is available at https://ddd.uab.cat/record/337343

under the terms of the CC BY-SA license.

PIPELINE DE DADES PER PRIORITZACIÓ DE VULNERABILITATS

Sara Martínez Llobregat

Resum—Aquest treball presenta el disseny i la implementació d'un pipeline de dades per prioritzar vulnerabilitats en una PIME simulada. La solució integra un inventari de 50 actius amb dades de NVD, EPSS i CISA KEV, normalitza el software instal·lat i el relaciona amb vulnerabilitats mitjançant identificadors CPE. Sobre les 17.692 deteccions obtingudes s'apliquen quatre models de scoring basats en CVSS, probabilitat d'explotació, explotació coneguda i context dels actius. La comparació mostra que el model contextual és el més adequat, ja que prioritza les vulnerabilitats amb més risc pràctic i impacte potencial sobre actius crítics o de producció. El resultat proporciona una metodologia reproduïble i justificable per orientar recursos de remediació limitats.

Paraules clau—Gestió de vulnerabilitats, priorització, pipeline de dades, CVSS, EPSS, CISA KEV, CPE, risc contextual.

Abstract—This project presents the design and implementation of a data pipeline for vulnerability prioritization in a simulated small and medium-sized enterprise. The solution integrates an inventory of 50 assets with data from NVD, EPSS, and CISA KEV, normalizes the installed software, and maps it to vulnerabilities through CPE identifiers. Four scoring models based on CVSS, exploitation probability, known exploitation, and asset context are applied to the 17,692 resulting detections. The comparison shows that the contextual model is the most suitable because it prioritizes vulnerabilities with greater practical risk and potential impact on critical or production assets. The resulting approach provides a reproducible and justifiable methodology for allocating limited remediation resources.

Index Terms—Vulnerability management, vulnerability prioritization, data pipeline, CVSS, EPSS, CISA KEV, CPE, contextual risk.

1 Introducció - Context del treball

1.1 Context i problemàtica

En els darrers anys, la gestió de vulnerabilitats s'ha convertit en un dels pilars fonamentals de la ciberseguretat empresarial. El creixement constant del nombre de vulnerabilitats publicades anualment, juntament amb l'augment d'atacs dirigits a organitzacions de tota mida, ha generat un escenari en què les empreses han de prendre decisions ràpides i informades sobre quines debilitats corregir primer. En aquest context, les petites i mitjanes empreses es troben especialment exposades, ja que sovint disposen de recursos limitats i no poden abordar totes les vulnerabilitats detectades de manera simultània. Això fa que la priorització esdevingui un element crític dins del procés de gestió de vulnerabilitats, més enllà de la simple identificació de les mateixes. [1], [3], [10]

Les vulnerabilitats conegudes són una de les principals causes de bretxes de seguretat. Molts incidents no provenen d'atacs desconeguts, sinó de vulnerabilitats ja publicades i documentades en bases de dades públiques com la NVD. Per tant, el problema no és la manca d'informació disponible, sinó la manca d'aplicació efectiva i d'una gestió adequada. Identificar una vulnerabilitat no implica necessàriament gestionar-la de manera eficient.

Diversos estudis reforcen aquesta problemàtica. Segons una investigació recent [2], el 68% de les organitzacions tarda més de 24 hores a corregir vulnerabilitats crítiques, i una de les principals dificultats és la manca de context o d'informació precisa per prioritzar correctament. Sovint, les puntuacions de risc s'analitzen de manera aïllada, sense integrar diferents fonts ni considerar el context real dels actius afectats, fet que dificulta determinar quines vulnerabilitats tenen realment més impacte sobre l'organització.

1.2 Motivació i plantejament del treball

En aquest escenari, es fa evident la necessitat d'un enfocament estructurat que permeti integrar fonts públiques de vulnerabilitats, contextualitzar-les dins d'un entorn empresarial i establir criteris de priorització basats en dades. Aquest treball proposa la creació d'un entorn que simula una PIME, sobre el qual s'integren diferents fonts d'informació de vulnerabilitats. L'objectiu és relacionar aquestes dades amb actius concrets i comparar diferents estratègies de scoring per determinar quina

permet una gestió més eficient i ajustada al risc real.

Per representar l'entorn empresarial, inicialment es va considerar la utilització d'una CMDB de codi obert. Tanmateix, aquesta opció introduïa una complexitat que no era imprescindible per assolir els objectius del treball. Per aquest motiu, es va optar per un inventari estructurat en format CSV, amb informació sobre els actius, el software instal·lat, les versions, els sistemes operatius, l'entorn i la criticitat. Aquesta alternativa permet representar de manera controlada i reproduïble els atributs necessaris d'una PIME simulada. [4], [5], [6], [7]

2 Objectius

2.1 Objectius generals

- Dissenyar i implementar un sistema de gestió i priorització de vulnerabilitats en un entorn empresarial simulat.
- Analitzar i comparar diferents estratègies de priorització per determinar quina s'ajusta millor.
- Integrar fonts públiques de vulnerabilitats en un sistema estructurat i contextualitzat.
- Aportar una metodologia objectiva i justificable per a la priorització de vulnerabilitats.

2.2 Objectius específics

- *OE1: Entorn i inventari*
  - OE1.1: Crear un entorn virtual que simuli una empresa a petita escala.
  - OE1.2: Definir un inventari estructurat d'actius (software, versions, servidors, etc.).
  - OE1.3: Emmagatzemar l'inventari en una base de dades pròpia.
- *OE2: Integració de dades*
  - OE2.1: Recollir dades de fonts públiques de vulnerabilitats (CVE/CVSS, EPSS, KEV).
  - OE2.2: Desenvolupar un pipeline ETL per descarregar, transformar i carregar les dades.
  - OE2.3: Normalitzar identificadors i estructures de dades.
  - OE2.4: Gestionar valors absents i possibles inconsistències.
- *OE3: Priorització*
  - OE3.1: Implementar com a mínim tres estratègies diferents de scoring/priorització.
  - OE3.2: Analitzar l'impacte de donar diferents pesos a cada font.
  - OE3.3: Comparar els resultats obtinguts amb cada estratègia.
- *OE4: Avaluació i resultats*
  - OE4.1: Generar una llista prioritzada de vulnerabilitats per a l'entorn simulat.
  - OE4.2: Justificar la priorització mitjançant mètriques i anàlisi comparativa.
  - OE4.3: Elaborar un informe d'avaluació amb conclusions.

3 Metodologia

3.1 Fases del treball

Per realitzar aquest projecte, s'ha fet una divisió en fases que s'organitzen temporalment. Aquesta planificació pot anar canviant segons l'avanç del projecte, i la que es presenta aquí reflecteix la versió final. Les sis fases en què està dividit el treball estan directament relacionades amb els diferents objectius plantejats.

Fase 1. Revisió teòrica i definició de l'abast: Es farà una revisió general dels coneixements que es considera que s'han d'aplicar en el projecte, així com de les diferents fonts que ja se sap que s'han d'utilitzar: CVE, CVSS, EPSS i CISA KEV, a més del concepte de CMDB. També es considera definir l'estructura del pipeline i de l'inventari d'actius de manera preliminar.

Fase 2. Definició de l'entorn i inventari d'actius (OE1): Es definirà l'estructura dels actius segons el mètode que s'hagi considerat (CMDB o inventari simplificat) i es crearà l'inventari inicial.

Fase 3. Obtenció i preparació de les dades de vulnerabilitats (OE2): Es descarregaran les dades de NVD, EPSS i CISA KEV, s'exploraran els datasets i es normalitzaran les estructures, obtenint així uns datasets preparats per a la integració.

Fase 4. Desenvolupament del pipeline de dades (OE2): S'implementarà el pipeline ETL, es normalitzaran els identificadors, es gestionaran els duplicats i es crearà la base de dades definitiva. D'aquesta manera, el sistema serà capaç d'unificar les fonts de vulnerabilitats.

Fase 5. Implementació de models de priorització (OE3): Es desenvoluparan diferents mètodes de scoring i s'aplicaran a l'inventari d'actius per tal de generar diverses llistes prioritzades. Així, s'obtindran resultats comparables entre els diferents models.

Fase 6. Anàlisi de resultats i redacció final (OE4): Es realitzarà una anàlisi comparativa i la interpretació de resultats d'aquesta, així com la redacció de les conclusions i de l'informe final.

Revisió: Aquest espai temporal es reserva per a la revisió global del treball, amb l'objectiu d'identificar possibles millores, especialment a nivell d'informe, i també per disposar de marge addicional en cas d'imprevistos durant el desenvolupament de les fases.

3.2 Planificació temporal

FASE	INICI	FINAL	DURACIÓ
Fase 1	08/03/2026	13/03/2026	5 dies
Fase 2	13/03/2026	27/03/2026	14 dies

Fase 3	27/03/2026	20/04/2026	24 dies
Fase 4	20/04/2026	20/05/2026	30 dies
Fase 5	20/05/2026	30/05/2026	10 dies
Fase 6	01/06/2026	14/06/2026	13 dies
Revisió	14/06/2026	28/06/2026	14 dies

Durant el desenvolupament, la metodologia s'ha ajustat a les particularitats de NVD, EPSS i CISA KEV. El canvi principal ha estat substituir el matching inicial per noms per una normalització controlada del software i un mapping específic cap a CPE, reduint així els falsos positius. També s'ha redissenyat la base de dades per eliminar redundàncies i separar les fonts, les deteccions i els resultats de scoring.

4 Estat de l'art

La gestió i priorització de vulnerabilitats ha evolucionat notablement durant els darrers anys a causa de l'augment constant de vulnerabilitats publicades i de la necessitat de reduir el risc de manera més eficient dins de les organitzacions. El gran volum d'informació disponible ha impulsat nous enfocaments orientats a una priorització basada en risc [8], [10], [11].

Una de les fonts més utilitzades són les puntuacions CVSS, que constitueixen un estàndard obert per avaluar la severitat tècnica de les vulnerabilitats. Aquest sistema assigna una puntuació numèrica i una classificació qualitativa per ajudar les organitzacions a prioritzar les accions de remediació. Tot i això, CVSS no representa per si sol el risc real, ja que no considera la probabilitat d'explotació ni el context de l'organització [8], [9].

Per aquest motiu, els enfocaments actuals combinen CVSS amb altres fonts. EPSS estima la probabilitat d'explotació d'una vulnerabilitat, mentre que CISA KEV identifica vulnerabilitats amb evidència d'explotació coneguda [13], [14]. També és important considerar el context dels actius, ja que la criticitat, l'entorn i l'exposició poden modificar el risc associat.

Aquesta evolució és especialment rellevant per a les PIME, que acostumen a disposar de recursos limitats i necessiten concentrar els esforços de remediació en les vulnerabilitats amb més risc pràctic. En aquest treball, aquest plantejament es concreta mitjançant la integració de CVSS, EPSS, CISA KEV i el context dels actius en diferents models de priorització.

5 Marc teòric

Aquest apartat presenta els conceptes, estàndards i tecnologies principals relacionats amb la gestió i priorització de vulnerabilitats utilitzats en el desenvolupament del projecte. També introdueix els elements necessaris per comprendre la integració de les fonts de dades i la contextualització dels actius dins d'un entorn empresarial.

5.1 Gestió i priorització de vulnerabilitats

La gestió de vulnerabilitats és el procés continu d'identificar, analitzar, prioritzar i corregir debilitats de seguretat. Com que les organitzacions no poden abordar totes les vulnerabilitats simultàniament, cal prioritzar-les segons la severitat, la probabilitat i evidència d'explotació i l'impacte potencial sobre els actius afectats [10].

5.2 Vulnerabilitats i estàndards

La gestió de vulnerabilitats utilitza diferents estàndards i fonts públiques per identificar, descriure i prioritzar vulnerabilitats. En aquest treball, aquests elements aporten informació complementària i es relacionen principalment mitjançant l'identificador CVE. [12], [13], [14]

Element	Funció dins del treball
CVE	Identificador únic que permet relacionar una vulnerabilitat entre les diferents fonts de dades.
NVD	Amplia la informació de les CVE amb descripcions, puntuacions CVSS i configuracions de productes afectats.
CVSS	Mesura la severitat tècnica d'una vulnerabilitat mitjançant una puntuació entre 0 i 10.
EPSS	Estima la probabilitat que una vulnerabilitat sigui explotada durant els propers 30 dies.
CISA KEV	Catàleg de vulnerabilitats amb evidència d'explotació activa coneguda.
CPE	Nomenclatura estandarditzada per identificar fabricants, productes i versions afectades.

CVSS proporciona un criteri inicial de severitat, però no representa per si sol la probabilitat d'explotació ni l'impacte de la vulnerabilitat dins d'una organització. Per aquest motiu, el projecte combina aquesta mètrica amb EPSS, CISA KEV i el context dels actius. Els identificadors CPE permeten relacionar el software de l'inventari amb les configuracions afectades descrites per NVD.

5.3 ETL i pipelines de dades en ciberseguretat

En entorns de ciberseguretat, la informació sobre vulnerabilitats prové de múltiples fonts i formats. Per integrar i analitzar aquestes dades s'utilitzen pipelines ETL (Extract, Transform, Load), que permeten obtenir, transformar i carregar la informació abans d'utilitzar-la.

En aquest treball, el pipeline integra dades procedents de NVD, EPSS i CISA KEV amb l'inventari d'actius de l'entorn simulat. Aquestes fonts proporcionen informació complementària i utilitzen formats diferents: NVD

publica les dades en JSON, mentre que EPSS, CISA KEV i l'inventari es processen a partir de fitxers CSV [12], [13], [14].

- **Extracció:** lectura dels feeds JSON de NVD i dels CSV d'EPSS, CISA KEV i l'inventari.
- **Transformació:** selecció de camps, conversió de tipus i tractament de valors absents, duplicats i registres no vàlids.
- **Normalització:** unificació dels noms del software i associació amb identificadors CPE.
- **Càrrega:** inserció de les dades processades en taules separades de SQL Server.
- **Validació:** comprovació de camps, recomptes i versions abans de generar les deteccions.

Aquest procés permet transformar dades heterogènies en una estructura comuna sobre la qual es poden detectar i prioritzar vulnerabilitats associades a actius concrets.

5.4 Inventari d'actius i CMDB

Un inventari d'actius recull informació sobre els sistemes d'una organització, com el software instal·lat, les versions, l'entorn i la criticitat. Aquest context permet diferenciar la prioritat d'una mateixa vulnerabilitat segons l'actiu afectat. Tot i que aquesta informació es pot gestionar mitjançant una CMDB, en aquest treball s'utilitza un inventari CSV simplificat, suficient per representar els atributs necessaris de l'entorn simulat [5], [6], [7].

6 Disseny de la solució

La solució proposada consisteix en el disseny d'un pipeline de dades per donar suport a la priorització de vulnerabilitats en un entorn empresarial simulat. El sistema parteix d'un inventari d'actius i l'enriqueix amb informació pública de vulnerabilitats procedent de diferents fonts.

6.1 Visió general de la solució

El pipeline es divideix en diferents fases, fet que facilita la revisió dels resultats intermedis i permet modificar una part del procés sense haver de reconstruir tota la solució.

Cada fase del pipeline té una responsabilitat concreta: carregar l'inventari, normalitzar el software instal·lat, ingerir les fonts externes, relacionar productes amb identificadors CPE, detectar vulnerabilitats en actius i provar models de priorització.

6.2 Arquitectura general del sistema

L'entrada de l'arquitectura consta de quatre blocs principals de dades: l'inventari d'actius simulat, els feeds de NVD, el dataset EPSS i el catàleg CISA KEV. A la sortida, el sistema genera taules de scoring, fitxers CSV i visualitzacions resum que permeten comparar diferents estratègies de priorització. [12], [13], [14]

El sistema es divideix en les capes següents:

- Capa d'inventari, formada pels actius simulats i el software instal·lat.
- Capa de normalització, encarregada de convertir noms de software en productes interns normalitzats.
- Capa de mapping, que relaciona els productes interns amb identificadors CPE utilitzats per NVD.
- Capa d'ingesta de fonts externes, que carrega NVD, EPSS i CISA KEV en taules separades.
- Capa de detecció, que creua l'inventari amb els productes afectats segons NVD i valida versions.
- Capa de scoring, que calcula diferents models de priorització.
- Capa de sortida, formada per CSVs i una visualització HTML per facilitar la interpretació dels resultats.

6.3 Disseny del pipeline de dades

El pipeline s'ha dissenyat seguint una lògica ETL, tot i que en alguns punts també incorpora fases d'enriquiment i validació. El procés comença amb l'extracció de dades des de fitxers CSV i JSON, continua amb la transformació i normalització dels camps, i finalitza amb la càrrega en una base de dades relacional.

En el cas de NVD, el pipeline llegeix feeds JSON 2.0 anuals corresponents al període 2020-2026. Aquests fitxers contenen informació de les CVEs, mètriques CVSS i configuracions CPE dels productes afectats. Durant la transformació, el sistema separa la informació en dues estructures: una taula de vulnerabilitats i una taula de productes afectats. Aquesta separació és necessària perquè una mateixa CVE pot afectar múltiples productes o versions. [12]

EPSS i CISA KEV es tracten com a fonts d'enriquiment. EPSS proporciona una puntuació probabilística associada a cada CVE, mentre que KEV indica si una vulnerabilitat forma part del catàleg de vulnerabilitats explotades conegudes. Aquestes fonts no substitueixen la informació de NVD, sinó que s'utilitzen posteriorment per millorar la priorització. [13], [14]

El pipeline també inclou una fase específica per a l'inventari. El camp de software dels actius pot contenir diversos productes separats en text, de manera que cal dividir-lo, identificar el producte corresponent i extreure'n la versió. Aquesta informació es carrega en una taula intermèdia de software instal·lat per actiu.

Finalment, el pipeline genera les deteccions de vulnerabilitats en actius. Aquesta fase creua el software instal·lat amb els mappings CPE i amb els productes afectats de NVD. Quan és possible, també comprova si la versió instal·lada entra dins el rang vulnerable indicat per NVD. El resultat és una taula que representa casos concrets de risc: una vulnerabilitat determinada que afecta un software instal·lat en un actiu concret.

6.4 Disseny de la base de dades i model de dades

La base de dades segueix un model relacional que separa

l'inventari, el catàleg de software, les fonts de vulnerabilitats, les deteccions i els resultats de priorització. Aquesta estructura permet conservar la traçabilitat de les dades i evitar duplicacions innecessàries.

Les taules assets, software_products i asset_software representen l'inventari normalitzat, mentre que product_cpe_mapping el relaciona amb els identificadors de NVD. Les fonts externes es mantenen a vulnerabilities, affected_products, epss_scores i kev_catalog. Finalment, asset_vulnerabilities conserva les deteccions i scoring_models i vulnerability_scores emmagatzemen els models i els resultats.

EPSS i KEV es mantenen en taules independents i es relacionen lògicament mitjançant cve_id, perquè tenen cobertures i ritmes d'actualització diferents. El model relacional complet es presenta a la Figura A2.1 de l'annex.

6.5 Disseny de l'inventari d'actius simulat

L'inventari d'actius s'ha dissenyat com una alternativa simplificada a una CMDB real. Inicialment es podia plantejar l'ús d'una eina completa de gestió de configuració, però aquesta opció hauria afegit una complexitat considerable que no era imprescindible per assolir els objectius del treball. Per aquest motiu, s'ha optat per un inventari estructurat en format CSV i carregat posteriorment a la base de dades.

L'inventari conté 50 actius simulats. Cada actiu disposa d'informació bàsica com el hostname, el tipus d'actiu, el sistema operatiu, l'entorn, la criticitat, el departament, l'adreça IP i el software instal·lat. Els actius es reparteixen entre servidors, bases de dades i endpoints, i també entre entorns de producció, desenvolupament i proves.

La presència dels camps `environment` i `criticality` és especialment important per al treball. Aquests atributs permeten incorporar context empresarial als models de scoring. Una mateixa vulnerabilitat pot tenir una prioritat diferent si afecta un actiu crític en producció o un actiu de baixa criticitat en un entorn de proves. Per tant, l'inventari no només serveix per saber quin software existeix, sinó també per interpretar la importància relativa de cada detecció.

El camp de software s'ha definit com una llista de productes amb versió. Aquesta decisió permet simular una situació més realista que un inventari sense versions, ja que en gestió de vulnerabilitats no n'hi ha prou amb saber que un producte està instal·lat: cal saber si la versió concreta entra dins del rang vulnerable.

6.6 Estratègia de normalització i matching software-CPE

Un dels punts més importants del disseny és la relació entre el software de l'inventari i els CPEs utilitzats per NVD. Aquesta relació no és trivial, perquè els noms dels productes en un inventari intern no sempre coincideixen exactament amb els noms normalitzats que apareixen a NVD.

Com els registres de la NVD no tenen noms estandarditzats, cal una exploració prèvia per veure l'estructura i com poden aparèixer els diferents productes que volem prioritzar. Intentar fer el matching directament pel fabricant o pel nom del producte podria provocar la pèrdua d'informació important. Per exemple, el vendor *apache* pot fer referència a productes molt diferents, com Apache HTTP Server, Tomcat, Traffic Server, NiFi o altres projectes.

Per evitar aquest problema, s'ha introduït una capa intermèdia de productes interns normalitzats. El sistema primer transforma el nom textual del software en un `product_key`, com ara `apache/http_server`, `apache/tomcat`, `openssl/openssl` o `google/chrome`. Després, aquest producte intern es relaciona amb un vendor i product CPE mitjançant una taula de mapping controlada.

Aquesta estratègia fa que el procés sigui més lent de preparar, però també més justificable. En lloc de confiar en coincidències massa genèriques, el sistema utilitza una correspondència explícita entre productes interns i identificadors CPE. Això redueix el risc de falsos positius i permet documentar millor les decisions preses.

La validació de versions és l'últim pas del matching. El sistema compara la versió instal·lada amb la versió exacta o els rangs vulnerables indicats per NVD.

6.7 Estratègies de scoring i priorització

La priorització s'ha dissenyat a partir de diversos models progressius. L'objectiu és comparar com canvia el resultat quan s'incorporen noves dimensions del risc. Els models parteixen d'un baseline senzill i afegeixen informació de manera incremental.

El primer model és *CVSS*, basat únicament en la severitat tècnica. Aquest model serveix com a punt de referència, ja que representa una forma habitual i simple de prioritzar vulnerabilitats.

El segon model és *CVSS_EPSS*, que combina CVSS amb EPSS. Amb aquest model, una severitat tècnica moderada amb una probabilitat d'explotació elevada pot guanyar prioritat respecte al baseline.

El tercer model és *CVSS_EPSS_KEV*, que afegeix la presència al catàleg CISA KEV. D'aquesta manera, el model permet destacar vulnerabilitats que ja tenen evidència d'explotació activa.

Finalment, el model CONTEXTUAL afegeix el context de l'actiu mitjançant una puntuació calculada a partir de la criticitat i l'entorn de l'actiu. Aquesta és l'estratègia més propera a una priorització orientada a risc empresarial.

Les variables es normalitzen abans de calcular el score. El CVSS es transforma a una escala 0-1 dividint-lo entre 10. EPSS ja es troba en una escala 0-1. La presència a KEV es representa com un valor binari. El context de l'actiu es calcula combinant la criticitat i l'entorn, donant més pes a la criticitat però considerant també si l'actiu es troba en

producció.

Els pesos inicials dels models són configurables i es guarden a la taula `scoring_models`. Això permet modificar-los o crear noves versions sense perdre la traçabilitat dels resultats. Els resultats calculats per cada model es guarden a `vulnerability_scores`, on cada detecció rep un score final, un ranking i un nivell de prioritat.

El flux complet del pipeline, des de la càrrega de l'inventari fins a la generació dels resultats prioritzats, es representa a la Figura A2.2 de l'annex.

7 IMPLEMENTACIÓ

7.1 Entorn i tecnologies utilitzades

La implementació del projecte s'ha desenvolupat principalment amb Python i SQL Server. Python s'ha utilitzat per construir els scripts de lectura, normalització, transformació i càrrega de dades, mentre que SQL Server s'ha utilitzat com a base de dades relacional per conservar les taules del pipeline i executar els creuaments entre fonts.

Les fonts de dades utilitzades tenen formats diferents. L'inventari d'actius, EPSS i CISA KEV es treballen a partir de fitxers CSV, mentre que NVD es processa a partir de feeds JSON 2.0 comprimits en format `.json.gz`. [12], [13], [14]

7.2 Implementació de l'inventari d'actius

L'inventari d'actius está incialment en format csv i conté 50 actius. La generació es fa amb un script, `inventari_actius.py`, que permet crear actius simulats amb una distribució realista de tipus d'actius, entorns i software instal·lat. Tot i això, per garantir que els resultats siguin reproduïbles, el pipeline actual parteix del CSV ja generat i versionat dins del repositori.

La càrrega de l'inventari a SQL Server es fa amb `assets_loader.py`. Aquest script llegeix el CSV, normalitza camps textuals bàsics i fa les comprovacions necessàries. Un cop preparat el DataFrame, insereix els registres a la taula `assets`. El fet de conservar l'inventari en un CSV facilita la reproducció del projecte. Si es volgués modificar l'escenari simulat, es podria actualitzar el fitxer d'entrada i tornar a executar la càrrega. Aquesta decisió també evita dependre d'una CMDB real, que hauria introduït una complexitat addicional fora de l'abast principal del treball.

7.3 Implementació de la ingesta de fonts externes

NVD

La ingesta de NVD es fa amb nvd_parser_v2.py. Aquest script substitueix el parser inicial, que estava pensat per processar un únic fitxer JSON i resultava insuficient per treballar amb un conjunt de dades més complet. En les primeres proves només s'havia utilitzat un fitxer parcial de NVD, fet que limitava molt la cobertura de vulnerabilitats i productes afectats. Per aquest motiu, es va desenvolupar una segona versió del parser, capaç de llegir els feeds anuals de NVD en format JSON 2.0, inclosos fitxers comprimits .json.gz. En l'estat actual del projecte, s'utilitzen els feeds corresponents al període 2020-2026. [12]

El parser recorre la clau `vulnerabilities` de cada feed i extreu la informació principal de cada CVE. Les dades normalitzades es carreguen a la taula `vulnerabilities.`

La selecció de CVSS es fa prioritzant les versions més recents. Per defecte, el parser exclou les CVEs amb estat *Rejected*, ja que no representen vulnerabilitats vàlides per al procés de priorització.

A més de les vulnerabilitats, el parser extreu les configuracions CPE afectades i les carrega a la taula `affected_products`. Aquesta taula conté el CPE complet, vendor, producte, versió i possibles rangs vulnerables. Aquesta separació és necessària perquè una mateixa CVE pot afectar molts productes o moltes versions diferents.

EPSS

La ingesta d'EPSS es fa amb `epss_parser.py`. El script llegeix el CSV principal i normalitza els noms de columna. Els valors numèrics es converteixen al tipus corresponent i s'eliminen duplicats per CVE. Les dades es carreguen a la taula `epss_scores`. [13]

CISA KEV

La ingesta de CISA KEV es fa amb `kev_parser.py`. El parser normalitza els noms de columna originals del CSV i fa les conversions necessàries. El resultat es carrega a la taula `kev_catalog`, que posteriorment es relaciona amb la resta del sistema mitjançant l'id del cve. [14]

7.4 Implementació de la base de dades

L'estructura de la base de dades es crea amb rebuild_schema.sql. Aquest script reconstrueix l'esquema complet del projecte i defineix les taules, claus primàries, claus foranes i índexs principals. La decisió de centralitzar la creació de l'esquema en un únic script SQL es va prendre després de detectar que alguns scripts Python havien acumulat lògica de creació i adaptació de taules. En la versió actual, els scripts Python assumeixen que l'esquema ja existeix i només s'encarreguen de carregar dades.

Les taules s'organitzen en blocs funcionals: inventari, catàleg de software, fonts de vulnerabilitats, deteccions i scoring. Les relacions principals es defineixen mitjançant claus foranes.

També s'han definit índexs sobre camps utilitzats habitualment en creuaments, com els ids. Aquests índexs són especialment importants després d'ampliar NVD als feeds 2020-2026, ja que la taula `affected_products` passa a tenir més d'un milió i mig de registres.

7.5 Normalització del software i mapping a CPE

La normalització del software instal·lat es fa amb `software_parser.py`. Aquest script separa els diferents productes indicats al camp software i intenta reconèixer cada producte a partir d'un conjunt de patrons definits al mateix script.

El primer resultat d'aquesta normalització és la taula `software_products`, que actua com a catàleg intern de productes.

El segon resultat és la taula `asset_software`, que relaciona cada actiu amb el software instal·lat i la versió extreta del nom original.

Finalment, el script genera la taula `product_cpe_mapping`. Aquesta taula relaciona cada producte intern amb un vendor i product CPE de NVD. Aquesta correspondència no es fa de manera automàtica a partir del vendor, sinó que es defineix de forma controlada per reduir falsos positius.

Aquest pas és essencial perquè NVD expressa els productes afectats mitjançant CPEs. Sense aquesta capa de mapping, el sistema no podria creuar de manera fiable el software de l'inventari amb les vulnerabilitats publicades.

7.6 Detecció de vulnerabilitats en actius

La detecció de vulnerabilitats en actius es fa amb asset_vulnerabilities.py. Aquest script creua el software instal·lat als actius amb els productes afectats de NVD, utilitzant com a punt d'unió el mapping entre els productes interns del sistema i els identificadors CPE.

El procés comença generant possibles candidats, és a dir, relacions potencials entre un software instal·lat i una CVE. Aquesta primera selecció es basa en la coincidència entre el CPE associat al producte intern i els CPEs afectats indicats per NVD. A més, la consulta incorpora informació de l'actiu i dades complementàries com CVSS, EPSS i KEV, que després s'utilitzen en la fase de priorització.

Un cop generats els candidats, es comprova si la versió instal·lada entra dins de la versió o rang de versions afectades. D'aquesta manera, no totes les coincidències per producte es consideren vulnerabilitats finals. En la càrrega definitiva només s'insereixen les deteccions que es consideren afectades o potencialment afectades, diferenciant entre coincidències exactes, coincidències dins de rang i casos en què només es pot confirmar el producte però no la versió amb prou precisió.

7.7 Implementació dels models de priorització

Els models de priorització s'han implementat amb `scoring.py.` Aquest script llegeix les deteccions de vulnerabilitats, les enriqueix amb informació de l'actiu, CVSS, EPSS i KEV, i calcula un score per cada model definit.

Un cop calculat el score, el sistema ordena les deteccions de cada model i classifica el resultat segons uns llindars inicials. Aquests llindars s'han definit com una primera aproximació i es poden ajustar si es vol calibrar millor la distribució de prioritats.

Els resultats es carreguen a la taula `vulnerability_scores` i també s'exporten com a csv. Els fitxers generats inclouen la llista completa de vulnerabilitats prioritzades, el top 20 per model, resums de scoring i distribució de prioritats.

7.8 Automatització, execució i validació

El pipeline s'executa mitjançant scripts independents en un ordre concret. En primer lloc, cal crear l'esquema de la base de dades, després es carreguen l'inventari, les fonts externes, la normalització del software, les deteccions i finalment els models de scoring.

La validació s'ha fet a diversos nivells. Un cop carregades les taules, s'han revisat recomptes, identificadors, nuls crítics i distribucions de valors. També s'ha comprovat que els identificadors de les taules amb identity comencin per 1 després de reconstruir o recarregar les dades.

8 RESULTATS I ANÀLISI

8.1 Resultats del pipeline

La càrrega final ha generat 199.734 vulnerabilitats a la taula `vulnerabilities` i 1.559.320 registres de productes afectats a `affected_products`. Aquest volum és molt superior al del primer conjunt NVD utilitzat, que contenia un subconjunt més reduït de CVEs. Aquesta ampliació ha estat important perquè ha permès obtenir una cobertura més realista sobre els productes presents a l'inventari. [12]

Pel que fa a les fonts d'enriquiment, s'han carregat 325.252 registres EPSS i 1.558 vulnerabilitats del catàleg CISA KEV. En el conjunt final de deteccions, 13.985 de les 17.692 deteccions tenen informació EPSS associada, i 290 deteccions corresponen a vulnerabilitats presents al catàleg KEV. [13], [14]

Com a resultat global, el pipeline ha generat 17.692 deteccions actiu-vulnerabilitat. Aquestes deteccions corresponen a 2.548 CVEs diferents i afecten els 50 actius de l'inventari simulat. Per tant, el sistema ja no es limita a carregar vulnerabilitats de manera aïllada, sinó que és capaç de relacionar-les amb actius concrets i preparar-les per a la priorització.

8.2 Vulnerabilitats detectades a l'entorn simulat

Les deteccions no es distribueixen de manera uniforme entre tots els productes. Els productes amb més deteccions són Google Chrome, MySQL, Apache Tomcat i Microsoft Office.

Els principals productes afectats són:

Producte	Deteccions	CVEs diferents	Actius afectats
Google Chrome	10.548	1.158	10
MySQL	2.220	508	6
Apache Tomcat	1.984	53	4
Microsoft Office	1.385	434	5
OpenSSL	404	55	7
MariaDB	352	77	5
Apache HTTP Server	280	57	5

A diferència de la primera execució amb un subconjunt NVD més petit, amb els feeds 2020-2026 apareixen deteccions per 17 productes diferents de l'inventari. Això indica que l'ampliació de la font NVD ha millorat clarament la cobertura del pipeline i ha permès obtenir resultats més representatius. [12]

També s'ha analitzat la distribució de deteccions segons el context dels actius. Els actius de producció acumulen 6.642 deteccions, mentre que els de desenvolupament en tenen 7.760 i els de proves 3.290. Pel que fa a la criticitat, els actius crítics tenen 3.167 deteccions i els actius d'alta criticitat en tenen 3.475. Aquestes dades són rellevants perquè mostren que una part important de les vulnerabilitats detectades afecta actius amb possible impacte empresarial elevat.

8.3 Resultats dels models de scoring

Sobre les 17.692 deteccions s'han aplicat quatre models de scoring, cada model calcula una puntuació per a cada detecció, de manera que s'han generat 70.768 scores en total.

El model *CVSS* classifica 6.886 deteccions com a crítiques i 4.826 com a altes. Aquesta distribució mostra que CVSS tendeix a concentrar moltes deteccions en nivells alts, ja que només considera la gravetat tècnica de la vulnerabilitat. [9]

El model *CVSS_EPSS* redueix el nombre de deteccions crítiques a 208, però manté 6.317 deteccions en nivell alt. Això passa perquè la probabilitat d'explotació modifica el pes de vulnerabilitats que, tot i tenir CVSS alt, no sempre tenen una probabilitat d'explotació igualment elevada. [13]

El model *CVSS_EPSS_KEV* dona més pes a vulnerabilitats amb explotació coneguda i situa aquest tipus de deteccions en posicions més altes del rànquing. [14]

Finalment, el model *CONTEXTUAL* afegeix la criticitat i l'entorn de l'actiu, de manera que prioritza no només la vulnerabilitat, sinó també el lloc on apareix dins l'organització simulada.

8.4 Comparativa entre models

La comparació entre models mostra diferències importants en les primeres posicions del rànquing. En el Top 20, el model *CVSS* inclou 15 deteccions KEV, mentre que els models *CVSS_EPSS*, *CVSS_EPSS_KEV* i *CONTEXTUAL* inclouen 20 deteccions KEV. Això indica que, quan s'afegeixen variables relacionades amb explotabilitat, les vulnerabilitats explotades conegudes passen a ocupar una posició més prioritària. [14]

També es veu una diferència clara en l'EPSS mitjà del Top 20. El model *CVSS* té un EPSS mitjà de 0,74, mentre que *CVSS_EPSS* i *CVSS_EPSS_KEV* arriben a 0,94. El model *CONTEXTUAL* manté un valor molt similar, amb un EPSS mitjà de 0,94. Això mostra que els models que incorporen EPSS seleccionen vulnerabilitats amb una probabilitat d'explotació més elevada. [13]

Model	KEV Top 20	EPSS mitjà	CVSS mitjà	High/ Critical	Product ion
1	15	0,74	9,88	13	13
2	20	0,94	9,85	11	11
3	20	0,94	9,85	11	11
4	20	0,94	9,67	17	17

El model contextual destaca perquè manté una alta presència de vulnerabilitats KEV i EPSS elevat, però al mateix temps augmenta el nombre de deteccions en actius de producció o d'alta criticitat. Per aquest motiu, és el model que millor reflecteix la idea de priorització basada en risc dins de l'entorn simulat. [13], [14]

8.5 Anàlisi de prioritats obtingudes

Les primeres posicions dels models mostren vulnerabilitats amb valors alts de CVSS, EPSS elevat i presència a KEV. Per exemple, apareixen CVEs associades a Redis, Apache HTTP Server i Apache Tomcat, algunes de les quals afecten actius de producció o actius classificats com a crítics o d'alta criticitat.

El model basat només en CVSS tendeix a prioritzar vulnerabilitats amb puntuació tècnica màxima, encara que no sempre siguin les més rellevants des del punt de vista de l'explotació o del context. En canvi, els models que incorporen EPSS i KEV eleven vulnerabilitats amb més evidència de risc pràctic. Això permet evitar que la priorització depengui exclusivament de la severitat tècnica. [8], [13], [14]

L'anàlisi de canvis de rànquing respecte al model CVSS també mostra casos interessants. Algunes vulnerabilitats amb CVSS mitjà, però amb EPSS molt elevat, pugen

moltes posicions en els models que incorporen explotabilitat. Aquest comportament és rellevant perquè reflecteix una situació habitual en gestió de vulnerabilitats: no sempre la vulnerabilitat amb més severitat tècnica és la que s'hauria d'atendre primer.

8.6 Limitacions dels resultats

Tot i que els resultats obtinguts són útils per comparar estratègies de priorització, cal tenir en compte diverses limitacions.

En primer lloc, la detecció depèn del mapping entre productes interns i CPEs de NVD. Tot i que s'ha definit una correspondència controlada, poden existir variants de producte o CPEs alternatius que no s'hagin inclòs. Això pot provocar que algunes vulnerabilitats no es detectin o que alguns productes tinguin menys cobertura de la que tindrien amb un mapping més complet.

També cal considerar la comparació de versions. El pipeline interpreta versions exactes i rangs indicats per NVD, però el versionat de software pot ser complex i no sempre segueix una estructura uniforme.

Finalment, els pesos dels models de scoring i els llindars de prioritat són una primera aproximació. S'han definit per permetre comparar models de manera clara, però podrien ajustar-se en funció de criteris organitzatius, tolerància al risc o polítiques internes de remediació. Per tant, els resultats no s'han d'interpretar com una classificació universal, sinó com una proposta de priorització dins de l'escenari simulat del treball.

9 Conclusions

9.1 Conclusions del treball

El treball ha permès dissenyar i implementar un pipeline funcional per prioritzar vulnerabilitats en un entorn empresarial simulat. La solució integra un inventari d'actius amb fonts públiques com NVD, EPSS i CISA KEV, normalitza la informació i genera deteccions de vulnerabilitats associades a actius concrets. [12], [13], [14]

Una de les aportacions principals és que el sistema no es limita a ordenar CVEs per severitat CVSS, sinó que incorpora altres factors com la probabilitat d'explotació, la presència al catàleg KEV i el context de l'actiu afectat. Això permet comparar diferents enfocaments de priorització i observar com canvien les decisions quan s'introdueix informació més propera al risc real. [8], [13], [14]

Els resultats obtinguts demostren que la incorporació progressiva d'aquestes variables modifica de manera significativa l'ordre de les vulnerabilitats. El model basat exclusivament en CVSS concentra un nombre elevat de deteccions en les categories de prioritat alta i crítica, mentre que els models que incorporen EPSS i KEV destaquen vulnerabilitats amb una probabilitat d'explotació elevada o amb evidència d'explotació coneguda. Finalment, el model CONTEXTUAL manté aquests criteris i afegeix la criticitat i l'entorn dels actius, fet que permet donar més prioritat a les deteccions que afecten sistemes rellevants per a l'organització.

A partir de la comparació realitzada, es conclou que el model CONTEXTUAL és el més adequat per a l'entorn de PIME plantejat. En el seu Top 20, totes les deteccions corresponen a vulnerabilitats presents a CISA KEV, 17 afecten actius de producció i 17 corresponen a actius de criticitat alta o crítica. D'aquesta manera, el model combina severitat tècnica, probabilitat i evidència d'explotació amb l'impacte potencial sobre l'organització.

Aquesta combinació és especialment rellevant per a una PIME, ja que la limitació de recursos impedeix abordar simultàniament totes les vulnerabilitats detectades. El model contextual permet concentrar els esforços de remediació en els casos amb més risc pràctic i més impacte empresarial. No obstant això, la seva aplicació depèn de disposar d'un inventari actualitzat i d'una classificació mínima dels actius. Quan aquesta informació no estigui disponible, el model que combina les tres fonts, CVSS, EPSS i KEV, pot constituir una alternativa inicial fins que l'organització disposi del context necessari.

9.2 Conclusions personals

El desenvolupament del projecte ha mostrat que la part més complexa no ha estat la lògica per implementar els scripts, sinó comprendre les diferents fonts de dades i prendre decisions de disseny coherents. El matching entre el software instal·lat i els CPE de NVD ha estat especialment rellevant, ja que una correspondència massa genèrica podia produir deteccions poc fiables. Això ha fet necessari explorar les dades, normalitzar els productes i definir mappings controlats abans de generar les deteccions.

També ha estat necessari adaptar la planificació inicial a mesura que es validaven les dades i els resultats intermedis. Algunes decisions es van modificar durant el desenvolupament, especialment l'estructura de la base de dades, el parser de NVD i l'estratègia de detecció. Tot i que aquests canvis van incrementar el temps dedicat a la implementació, van permetre obtenir un resultat final més coherent, traçable i robust.

Un dels principals aprenentatges del treball ha estat comprovar que la qualitat de la priorització no depèn únicament de la fórmula de scoring. També està condicionada per la qualitat de l'inventari, la normalització del software, el mapping amb els CPE i la validació de les versions afectades. Per tant, disposar d'un model de priorització complet no és suficient si les dades d'entrada o les relacions entre les fonts no són fiables.

9.3 Objectius assolits

L'OE1, relacionat amb la definició de l'entorn empresarial

simulat, s'ha completat amb la creació d'un inventari estructurat de 50 actius. Aquest inventari inclou informació sobre el software i les versions instal·lades, el sistema operatiu, l'entorn i la criticitat de cada actiu. Les dades s'han carregat en una base de dades relacional i s'han utilitzat com a context empresarial per a la detecció i la priorització.

L'OE2, centrat en la integració de dades, s'ha assolit mitjançant el desenvolupament d'un pipeline capaç de processar i normalitzar dades de NVD, EPSS i CISA KEV. El sistema integra fonts amb formats i estructures diferents, normalitza el software de l'inventari, el relaciona amb identificadors CPE i genera deteccions de vulnerabilitats sobre actius concrets. En total, el procés ha tractat 199.734 vulnerabilitats i 1.559.320 registres de productes afectats, i ha generat 17.692 deteccions actiu-vulnerabilitat.

L'OE3, corresponent a la priorització, s'ha completat amb la implementació de quatre models: CVSS, CVSS_EPSS, CVSS_EPSS_KEV i CONTEXTUAL. Aquests models han permès estudiar progressivament l'impacte d'incorporar la probabilitat d'explotació, l'evidència d'explotació coneguda i el context empresarial. En total, s'han calculat 70.768 puntuacions i s'han comparat els rànquings i les distribucions de prioritat obtinguts.

Finalment, l'OE4, vinculat a l'avaluació dels resultats, s'ha assolit mitjançant l'anàlisi comparativa dels quatre models i la selecció d'un enfocament final. Els resultats mostren que el model CONTEXTUAL és el més adequat per a l'entorn de PIME plantejat, ja que combina els indicadors tècnics i d'explotació amb la criticitat i l'entorn dels actius. Aquesta combinació permet dirigir els recursos limitats de remediació cap a les vulnerabilitats amb més risc pràctic i més impacte potencial sobre l'organització.

Per tant, s'han complert els objectius plantejats: s'ha construït un entorn simulat, s'han integrat i normalitzat les fonts de dades, s'han desenvolupat diferents estratègies de priorització i s'ha justificat la selecció d'un model final a partir dels resultats obtinguts.

9.4 Possibles millores i treball futur

Com a treball futur, es podria provar el sistema sobre un petit parc de màquines virtuals amb actius reals, en lloc de treballar únicament amb un inventari simulat. També es podria ampliar perquè permetés realitzar consultes específiques per actiu, producte o criticitat, i no només generar rànquings globals.

Altres millores serien afinar la validació de versions, automatitzar l'actualització periòdica de les fonts i incorporar informació addicional, com l'exposició dels actius o els controls de seguretat existents.

Finalment, seria especialment rellevant validar el model CONTEXTUAL en una PIME real. Aquesta validació permetria ajustar els pesos de CVSS, EPSS, KEV i del context, així com adaptar els llindars de prioritat a la tolerància al risc i als recursos de remediació disponibles.

Agraïments

Vull agrair al meu tutor, Enric Alibech, l'acompanyament i l'ajuda proporcionada durant el desenvolupament del treball. També vull donar les gràcies a la meva família i als meus amics pel seu suport i per motivar-me en els moments en què més em costava avançar.

Bibliografia

[1] Deloitte. Estado de la ciberseguridad.
https://www.deloitte.com/es/es/services/consulting/research/estado-ciberseguridad.html
[2] Swimlane. Investigación sobre gestión de vulnerabilidades.
https://swimlane.com/es/noticias/investigacion-sobre-gestion-de-vulnerabilidades/
[3] Gesprodat. El riesgo silencioso tras el cierre del año: parches pendientes y vulnerabilidades acumuladas.
https://gesprodat.com/actualidad/el-riesgo-silencioso-tras-el-cierre-del-ano-parches-pendientes-y-vulnerabilidades-acumuladas/
[4] Galtier. Inventario de activos fijos.
https://galtier.es/inventario-activos-fijos/
[5] Hackmetrix. Inventario de activos en seguridad de la información.
https://blog.hackmetrix.com/inventario-de-activos-seguridad-de-la-informacion/
[6] InvGate. IT asset inventory.
https://invgate.com/es/itsm/it-asset-management/it-asset-inventory
[7] CloudQuery. Asset inventory examples and best practices.
https://www.cloudquery.io/learning-center/asset-inventory-examples-best-practices
[8] Picus Security. Vulnerability prioritization: Why CVSS isn't enough.
https://www.picussecurity.com/resource/blog/vulnerability-prioritization-why-cvss-isnt-enough
[9] INCIBE. CVSS 3.0.
https://www.incibe.es/incibe-cert/blog/cvss3-0
[10] TuxCare. Gestión de vulnerabilidades.
https://tuxcare.com/es/blog/vulnerabilities-management/
[11] Red Seguridad. La gestión de vulnerabilidades evoluciona hacia un modelo continuo y basado en la exposición.
https://www.redseguridad.com/actualidad/la-gestion-de-vulnerabilidades-evoluciona-hacia-un-modelo-continuo-y-basado-en-la-exposicion_20260310.html
[12] National Institute of Standards and Technology. National Vulnerability Database data feeds.
https://nvd.nist.gov/vuln/data-feeds
[13] FIRST. EPSS data and statistics.
https://www.first.org/epss/data_stats
[14] Cybersecurity and Infrastructure Security Agency. Known exploited vulnerabilities catalog.
https://www.cisa.gov/known-exploited-vulnerabilities-catalog

APÈNDIX

A1. ÚS D'EINES D'INTEL·LIGÈNCIA ARTIFICIAL

Durant el desenvolupament del treball s'han utilitzat eines d'intel·ligència artificial com a suport puntual, principalment per ajudar en la implementació de scripts en Python i per contrastar informació tècnica concreta.

Aquest ús s'ha plantejat com una eina auxiliar i no com un substitut del procés de disseny i raonament del projecte. Les decisions relacionades amb l'arquitectura del pipeline, la integració de dades, el matching entre productes i la definició dels criteris de priorització s'han treballat i validat de manera pròpia.

A2. ARQUITECTURA I MODEL DE DADES

La solució s'ha implementat mitjançant una arquitectura modular que separa l'inventari d'actius, la normalització del software, les fonts de vulnerabilitats, la detecció i la priorització. Aquesta separació permet conservar la traçabilitat de les dades i revisar els resultats intermedis de cada fase del pipeline.

La Figura A2.1 presenta una versió simplificada del model relacional. Les línies contínues representen relacions implementades mitjançant claus foranes. Les línies discontínues indiquen les relacions lògiques d'EPSS i CISA KEV amb vulnerabilities mitjançant cve_id. Finalment, la línia puntejada representa l'ús del mapping CPE durant el procés de detecció, sense constituir una clau forana.

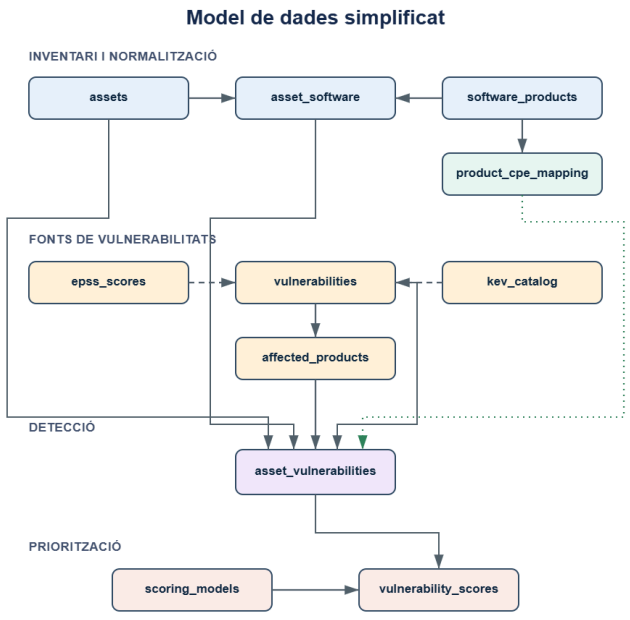

Figura A2.1 Model de dades simplificat del pipeline.

El flux de processament comença amb l'inventari d'actius i la normalització del software instal·lat. Els productes identificats es relacionen amb identificadors CPE, que permeten creuar-los amb les vulnerabilitats i els productes afectats de NVD. Les deteccions resultants s'enriqueixen amb EPSS i CISA KEV i, finalment, s'apliquen els diferents models de scoring.

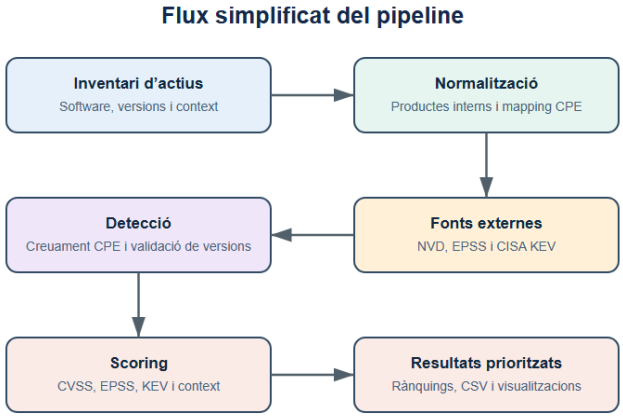

Figura A2.2 Flux simplificat del pipeline de dades, des de l'inventari d'actius fins a la generació dels resultats prioritzats.

El repositori lliurat conté els scripts d'ingesta, normalització, detecció i scoring, juntament amb els fitxers de dades i les sortides completes generades durant l'execució del pipeline.

A3. FRAGMENTS CLAU DE LA IMPLEMENTACIÓ

Aquest annex presenta una selecció dels fragments de codi més representatius del projecte: la generació de l'inventari d'actius, el matching entre software i CPE, i la definició i el càlcul dels models de priorització. Els fragments s'han simplificat per facilitar-ne la lectura; els scripts complets es troben al repositori lliurat.

A3.1 GENERACIÓ DE L'INVENTARI D'ACTIUS

L'inventari simulat es genera combinant distribucions de tipus d'actiu i entorn amb sistemes operatius, productes, versions i nivells de criticitat. Cada registre es valida abans d'incorporar-lo al CSV definitiu de 50 actius.

```
def generate_asset(asset_id):
    asset_type = weighted_choice(ASSET_DISTRIBUTION)
    env_prefix = weighted_choice(ENV_DISTRIBUTION)

    environment = get_environment(env_prefix)
    hostname = generate_hostname(asset_type,
env_prefix)
    os = generate_os(asset_type)
    software_list = generate_software(asset_type)
    software = ";".join(software_list)

    criticality = get_criticality(environment)
```

```
    ip_address = generate_ip()
    department = random.choice(DEPARTMENTS)

    return {
        "asset_id": asset_id,
        "hostname": hostname,
        "asset_type": asset_type,
        "os": os,
        "environment": environment,
        "criticality": criticality,
        "software": software,
        "ip_address": ip_address,
        "department": department
    }

assets = []
for i in range(1, NUM_ASSETS + 1):
    while True:
        asset = generate_asset(i)
        if validate_asset(asset):
            assets.append(asset)
            break

output_path = Path(__file__).resolve().parent /
"assets.csv"

with open(output_path, "w", newline="",
encoding="utf-8") as f:
    writer = csv.DictWriter(f,
fieldnames=assets[0].keys())
    writer.writeheader()
    writer.writerows(assets)
```

Fragment A3.1 Generació, validació i exportació de l'inventari simulat.

A3.2 Matching entre software i CPE

Els noms de software de l'inventari es transformen en productes interns normalitzats. Posteriorment, aquests productes es relacionen de manera controlada amb els valors de fabricant i producte utilitzats pels CPE de NVD.

```
PRODUCT_CPE_MAPPINGS = [
    ("apache/http_server", "apache", "http_server",
     "manual_verified", 1.00, None),

    ("apache/tomcat", "apache", "tomcat",
     "manual_verified", 1.00, None),

    ("f5/nginx_open_source", "f5", "nginx_open_source",
     "manual_verified", 0.95, "NVD utilitza F5 com a
vendor."),

    ("vmware/rabbitmq", "vmware", "rabbitmq",
     "manual_candidate", 0.85,
     "RabbitMQ ha tingut diferents vendors en CPE."),
]
```

Fragment A3.2 Exemple de mappings controlats entre productes interns i CPE.

Aquest plantejament permet diferenciar productes que comparteixen fabricant, com Apache HTTP Server i Apache Tomcat, i assignar una confiança inferior als mappings amb més incertesa. D'aquesta manera, es redueix el risc d'associar vulnerabilitats a productes incorrectes.

A3.3 Definició dels models de priorització

Els quatre models es defineixen mitjançant pesos configurables. Cada model incorpora progressivament noves dimensions del risc: severitat tècnica, probabilitat d'explotació, explotació coneguda i context de l'actiu.

```
SCORING_MODELS = [
    {
        "model_name": "CVSS",
        "description": "Baseline basat nomes en
severitat tecnica CVSS.",
        "cvss_weight": 1.00,
        "epss_weight": 0.00,
        "kev_weight": 0.00,
        "criticality_weight": 0.00,
    },
    {
        "model_name": "CVSS_EPSS",
        "description": "Model que combina severitat
tecnica CVSS i probabilitat d'explotacio EPSS.",
        "cvss_weight": 0.70,
        "epss_weight": 0.30,
        "kev_weight": 0.00,
        "criticality_weight": 0.00,
    },
    {
        "model_name": "CVSS_EPSS_KEV",
        "description": "Model que afegeix explotacio
coneguda segons CISA KEV.",
        "cvss_weight": 0.60,
        "epss_weight": 0.25,
        "kev_weight": 0.15,
        "criticality_weight": 0.00,
    },
    {
        "model_name": "CONTEXTUAL",
        "description": "Model complet amb CVSS, EPSS,
KEV i context de l'actiu.",
        "cvss_weight": 0.45,
        "epss_weight": 0.25,
        "kev_weight": 0.15,
        "criticality_weight": 0.15,
    },
```

```
]

CRITICALITY_VALUES = {
    "low": 0.25,
    "medium": 0.50,
    "high": 0.75,
    "critical": 1.00,
}

ENVIRONMENT_VALUES = {
    "testing": 0.30,
    "development": 0.40,
    "production": 1.00,
}
```

Fragment A3.3 Resum simplificat dels pesos dels models implementats.

Abans del càlcul, totes les variables es normalitzen en una escala entre 0 i 1. En el model contextual, la criticitat representa el 70 % del valor contextual i l'entorn de l'actiu el 30 %. El score final es calcula com una suma ponderada i es transforma a una escala de 0 a 100.

```
scoring_df["context_normalized"] = (
    0.70 * scoring_df["criticality_normalized"]
    + 0.30 * scoring_df["environment_normalized"]
)

model_df["final_score"] = 100 * (
    model["cvss_weight"] * model_df["cvss_normalized"]
    + model["epss_weight"] *
model_df["epss_normalized"]
    + model["kev_weight"] * model_df["kev_normalized"]
    + model["criticality_weight"]
      * model_df["context_normalized"]
)
model_df = model_df.sort_values(
    "final_score", ascending=False
)
model_df["priority_rank"] = range(1, len(model_df) + 1)
```

Fragment A3.4 Càlcul del score final i generació del rànquing.

Aquesta selecció mostra les decisions principals de la implementació sense reproduir els scripts complets. El repositori conté també els parsers de NVD, EPSS i CISA KEV, la creació de l'esquema SQL, la detecció de vulnerabilitats i la generació de les sortides d'anàlisi.

A4. AMPLIACIÓ DELS RESULTATS

Aquest annex amplia la comparació presentada a l'apartat de resultats. La Figura A3 mostra com els quatre models distribueixen les 17.692 deteccions entre els nivells de prioritat.

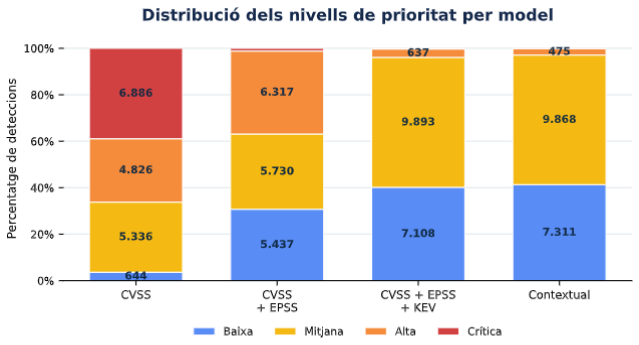

Figura A4. Distribució dels nivells de prioritat obtinguts amb cadascun dels models.

El model basat exclusivament en CVSS concentra 11.712 deteccions en els nivells alt o crític. En canvi, els models que incorporen EPSS, KEV i context redistribueixen les deteccions i generen una classificació més selectiva. El model contextual classifica 38 deteccions com a crítiques i 475 com a altes, fet que permet reduir el conjunt que requereix una actuació més immediata.

La taula següent mostra exemples de deteccions que pugen de manera significativa respecte al rànquing CVSS:

Model	CVE	Motiu principal	Canvi de rànquing
CVSS + EPSS	CVE-2022-3590	EPSS molt elevat (0,912)	11.715 → 570
CVSS + EPSS + KEV	CVE-2023-36761	Presència al catàleg KEV	10.105 → 1.340
CVSS + EPSS + KEV	CVE-2023-44487	KEV i EPSS molt elevat (0,944)	7.951 → 41
Contextual	CVE-2022-36021	Actiu crític en entorn de producció	12.026 → 1.008

Taula A1. Exemples de deteccions amb les pujades de rànquing més representatives respecte al model CVSS.

Els exemples mostren que una puntuació CVSS moderada no implica necessàriament una prioritat baixa. Un EPSS elevat pot fer pujar deteccions amb probabilitat d'explotació significativa, mentre que la presència a CISA KEV destaca vulnerabilitats amb evidència d'explotació coneguda, fins i tot quan l'EPSS és reduït. Finalment, el model contextual incrementa la prioritat de deteccions situades en actius crítics o de producció. Aquest comportament reforça la selecció del model contextual com l'opció més adequada per orientar els recursos limitats d'una PIME.