Convocatoria Colaboración Público-Privada CPP2021-009025

DICYME:

Dynamic Industrial CYberrisk Modelling based on Evidence

Iniciativa Conjunta de:

DENEXUS

Universidad Rey Juan Carlos

ENTREGABLE 1.3:

Prototipo final del sistema de recogida de evidencias (integrando datos internos y externos)

Coordinadores:

Romy R. Ravines (DeNexus Tech)
Isaac Martín de Diego (Universidad Rey Juan Carlos)
Alberto Fernández Isabel (Universidad Rey Juan Carlos)

MINISTERIO DE CIENCIA E INNOVACIÓN

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

AGENCIA ESTATAL DE INVESTIGACIÓN

Contenido

1 INTRODUCCIÓN Y OBJETIVOS

Este documento resume los aspectos más importantes de diseño e implementación del entregable 1.3 del Proyecto, es decir, del prototipo final del sistema de recogida de evidencias, integrando datos internos y externos. Este sistema consolida las soluciones desarrolladas en las tareas 1.1 y 1.2 del Proyecto, dentro de la Actividad 1 "Sistema de recogida de evidencias" e integra nuevas fuentes de datos internas y externas, con el objetivo de proporcionar una visión integral del ciber riesgo industrial. El ETL se ha diseñado para alimentar los modelos de probabilidad e impacto desarrollados en actividades relacionadas, proporcionando un flujo continuo de evidencias para la toma de decisiones.

En las siguientes secciones se explica:

- Qué tipo de datos se van a utilizar y cómo se transforman en información útil como entrada para el cálculo de probabilidades o impactos.
- Qué flujos de evidencias de ciber riesgo provenientes de diferentes fuentes externas relacionadas con las organizaciones y el contexto de ciber amenazas se han tenido en cuenta.
- Qué tipo de transformaciones de estos datos se han llevado a cabo para conseguir información útil para el modelado del ciber riesgo industrial.
- Qué datos y código se encuentran disponibles en este entregable (*deriskGroup / DICyME Project · GitLab*) y cómo pueden emplearse.

2 ALCANCE DEL SISTEMA ETL

2.1 Fuentes de datos internas

Como descrito en el entregable E.1.1, en este proyecto se muestra el tipo de información que proporciona un IDS (*Intrusion Detection System*) y los indicadores que se pueden construir y monitorizar. Los IDS proporcionan información sobre los dispositivos y tráfico en las redes internas, por lo tanto, se trata de información muy confidencial que DeNexus™ no puede compartir. Por ese motivo, los datos internos usados en este proyecto corresponden a una muestra anonimizada de datos reales que sirven para ilustrar los indicadores propuestos y su uso en la cuantificación del ciber riesgo.

DeNexus trabaja con los principales sistemas de IDS del mercado: *Nozomi*, *Forescout*, *Claroty*, y *Tenable*. Los procesos de adquisición de datos hacen uso de las API proporcionados por los fabricantes de dicha tecnología. Los procesos de adquisición y construcción de indicadores son parte de DeRISK™. Por confidencialidad no se pueden describir en este documento.

2.2 Fuentes de datos externas

Las fuentes internas comprenden datos recopilados de sistemas de seguridad. El sistema ETL también incluye fuentes externas:

- **Ciber incidentes:** Bases de datos como *ICS Strive*, *CISSM* y *EuroRepoC* que registran incidentes de ciberseguridad relevantes, incluidos ya en el entregable E1.2.
- **Actores de amenazas:** De la base de datos pública *Electronic Transactions Development Agency*, que estructura información como los nombres asociados a los actores, motivación, sectores y países objetivo, herramientas y *software* que emplean, etc., tal y como se describe también en el entregable E1.2.
- **Datos firmográficos:** Datos de *RocketReach*[1] y *CompaniesMarketCap*, que proporcionan información sobre tamaño, ubicación y capitalización de empresas.
- **Menciones en medios:** Utilizando *Determ*[2] para estimar reputación corporativa en medios de información online.
- **Victimización**: Indicadores que determinan cómo de apetecible es una compañía para un actor de amenazas. En concreto se obtiene la cantidad de dispositivos visibles en Internet desde *Shodan*, y la cantidad de apariciones en *leaks* de información de *ransomware*, de *Ransomware.live*.
- **Tácticas y técnicas de MITRE ATT&CK**: Dentro de este marco ampliamente utilizado para describir las acciones de los actores de amenazas durante un ciberataque, las tácticas representan los objetivos estratégicos de un atacante en cada fase del ataque, mientras que las técnicas detallan los métodos específicos utilizados para lograr esos objetivos.
- **Common Weakness Enumeration** (**CWE**): Clasificación de debilidades en software y hardware, descritas con un identificador único, detalles técnicos y ejemplos de explotación.

La Tabla 1. Fuentes de datos externas recoge las diferentes fuentes de datos, así como los métodos empleados para su obtención e información adicional. Dentro de los métodos de extracción, además de categorías como el uso de APIs, se han diferenciado dos tipos de *scraping web*: las peticiones a los *endpoints* o URLs desde el código, y la navegación automatizada a los sitios web a través del navegador, empleado en uno de los casos para obtener información generada en tiempo real con JavaScript.

2.3 Propósito del ETL

El ETL sirve para extraer, unificar, normalizar y cargar datos de múltiples fuentes en un sistema centralizado que alimenta los modelos desarrollados en DICYME. Esto incluye cálculos dinámicos sobre probabilidad e impacto de riesgos cibernéticos.

[1] Durante el desarrollo de este proyecto se usa la suscripción activa de DeNexus en *RocketReach*. Existen muchas otras soluciones que proporcionan la misma información.

[2] Durante el desarrollo de este proyecto se usa la suscripción activa de DeNexus en *Determ*. Existen muchas otras soluciones que proporcionan la misma información

Fuente	Tipo de datos	Extracción	Protección de los datos	Autenticación
ICS Strive, CISSM y Kon Briefing	Ciber incidentes	Peticiones a los *endpoints*	Información pública abierta	No aplica
Electronic Transactions Development Agency	Actores de amenazas	Peticiones a los *endpoints*	Información pública abierta	No aplica
CompaniesMarketCap	Firmográficos	Peticiones a los *endpoints*	Información pública abierta	No aplica
Ransomware.live	*Leaks* de *ransomware*	Peticiones a los *endpoints*	Información pública abierta	No aplica
RocketReach	Firmográficos	Peticiones a los *endpoints*	Suscripción costeada por DeNexus	Cookies de sesión de la web tras autenticarse con credenciales
Determ	Menciones en medios	Navegación automatizada	Suscripción costeada por DeNexus	Credenciales
Shodan	Dispositivos conectados a Internet	API	Suscripción gratuita de investigadores de la URJC	Token API
MITRE ATT&CK	Tácticas y técnicas de actores	Peticiones a los *endpoints*	Información pública abierta	No aplica
Common Weakness Enumeration (CWE)	Descripción de Vulnerabilidades	Peticiones a los *endpoints*	Información pública abierta	No aplica

Tabla 1. Fuentes de datos externas

3 ARQUITECTURA DEL PROTOTIPO

3.1 Estructura general

La ETL se basa en una arquitectura que combina procesamiento distribuido y orquestación centralizada. Las tres etapas principales son:

3.1.1 Extracción

En esta fase, se recopilan datos desde múltiples fuentes internas y externas, empleando scripts especializados que se ajustan a las características específicas de cada fuente:

- **Fuentes internas**: Como se comenta arriba, se usan las APIs proporcionadas por los fabricantes. Las integraciones han sido desarrolladas por DeNexus™ como parte del producto DeRISK. DeNexus ha puesto a disposición ejemplos de estos datos para el desarrollo del proyecto. Los códigos fuente son confidenciales y propiedad de DeNexus.

- **Fuentes externas**: Scripts como *rocketreach.py* y *companiesmarketcap.py* se conectan con los endpoints de las herramientas homónimas para obtener datos firmográficos. También lo hace *criticInfo.py* para obtener los *leaks* de *ransomware*, así como los algoritmos de extracción de información de ciber incidentes o el acceso a la base de datos de MITRE ATT&CK.

  Por otro lado, *get_determ_raw.py* utiliza técnicas de navegación automatizada para obtener datos de menciones en medios de Determ. También se usan interfaces de programación de aplicaciones (*Application Programming Interfaces*, APIs) para obtener los datos de dispositivos desde Shodan en *dorch.py*.

Para garantizar la eficiencia y la seguridad durante la extracción, se implementaron:

- **Autenticación segura**: Uso de *tokens* de acceso API y credenciales y cookies de sesión para garantizar la correcta autenticación con las cuentas y suscripciones disponibles.
- **Paralelización**: Extracciones simultáneas de diferentes fuentes para reducir los tiempos de procesamiento.

3.1.2 Transformación

Una vez extraídos, los datos pasan por procesos de limpieza, normalización y enriquecimiento para ser uniformes y aptos para el análisis. Esto incluye:

- **Normalización de formatos**: Conversión a estructuras JSON y CSV estándar, asegurando su estructuración y compatibilidad con las herramientas de modelado.
- **Combinación de fuentes y enriquecimiento**: Se obtienen datos complementarios de varias fuentes que ayudan a enriquecer el conjunto de datos final, así como completar datos faltantes en alguna de las fuentes. Este proceso es el realizado, entre otros, con *RocketReach* y *CompaniesMarketCap*, cuyos datos son combinados, así como los ciber incidentes. No obstante, aunque estos son combinados en una tabla común en la medida de lo posible (pues cada base de datos proporciona diferentes variables), aún no se produce enriquecimiento entre las diferentes fuentes, pues ha quedado como trabajo futuro la limpieza de duplicados y combinación dado el arduo esfuerzo que requiere.
- **Validación de datos**: Se incluyen controles y se implementan soluciones para detectar anomalías, datos faltantes o inconsistencias en tiempo real, permitiendo su corrección antes de la carga.

3.1.3 Carga

La etapa final consiste en almacenar los datos procesados en una base de datos centralizada. Este almacenamiento se realiza en un formato optimizado para consultas rápidas y análisis eficiente:

- **Bases de datos utilizadas**: GitHub para almacenamiento compartido inicial de las fuentes de datos de atractivo y MongoDB para datos semiestructurados en servidores propios de la Universidad Rey Juan Carlos.

- **Optimización para análisis**: Indexación de campos clave como vulnerabilidades, popularidad y atractivo.
- **Compatibilidad con modelos**: Los datos se cargan en formatos compatibles con los modelos de probabilidad e impacto, incluyendo exportaciones en JSON y CSV para sistemas externos.

3.2 Flujo de trabajo de ETL de AT2ATK (*Atractivo)*

La extracción, transformación y carga de los datos firmográficos, menciones en medios y victimización de atractivo se realiza siguiendo el esquema de la Ilustración 1. Flujo del ETL para atractivo, orquestado a través de *GitHub Workflows* y creando varios conjuntos de datos intermedios. Tras la búsqueda de nuevos incidentes de ciberseguridad, se desencadena manualmente el flujo de búsqueda de datos. Este, detallado en la Ilustración 2. *GitHub Workflow* de extracción de datos, orquesta la extracción de los tres tipos de datos. Es preciso recordar que, para este modelo, se extraen datos tanto de incidentes como de no incidentes. Este mismo *Workflow* gestiona la obtención de no incidentes desde *RocketReach* y, en caso de existir, extrae también sus datos (véanse los entregables E2.1 y E2.2, Primer y segundo prototipos del módulo de medida/estimación de probabilidad e impacto).

Con los datos firmográficos tanto de *RocketReach* como de *CompaniesMarketCap* se crea *dataset_firmographics_raw.csv*. Este se ordena todas las noches para facilitar un posterior análisis y búsqueda manual de los datos. Por otro lado, se procesa para combinar ambas fuentes, dando lugar a dataset_firmographics.csv.

Para la reputación online, los archivos *.csv* que proporciona *Determ* con los datos sin procesar de las menciones se almacenan en el directorio *raw_data/*. Seguidamente, se procesan para generar el conjunto de datos dataset_determ.csv con el cálculo de la reputación online de cada entidad (véase el entregable E1.4 Documentación de las técnicas y transformaciones de datos en información para su consumo en modelos de ciber riesgo OT).

Como último paso de la extracción, la cantidad de dispositivos visibles en Internet y los *leaks* de *ransomware* de cada entidad se almacenan en *dataset_victimization.csv*.

Una vez se han obtenido todos los datos, de manera periódica cada noche se combinan los tres conjuntos de datos parciales en *dataset_merged.csv*, que contiene los datos de las tres características de atractivo para ser utilizados por los modelos. Este conjunto de datos se vuelca a la base de datos MongoDB mantenida por el equipo de la Universidad Rey Juan Carlos en sus servidores, que sirve como fuente a la herramienta de visualización y soporte a la toma de decisiones.

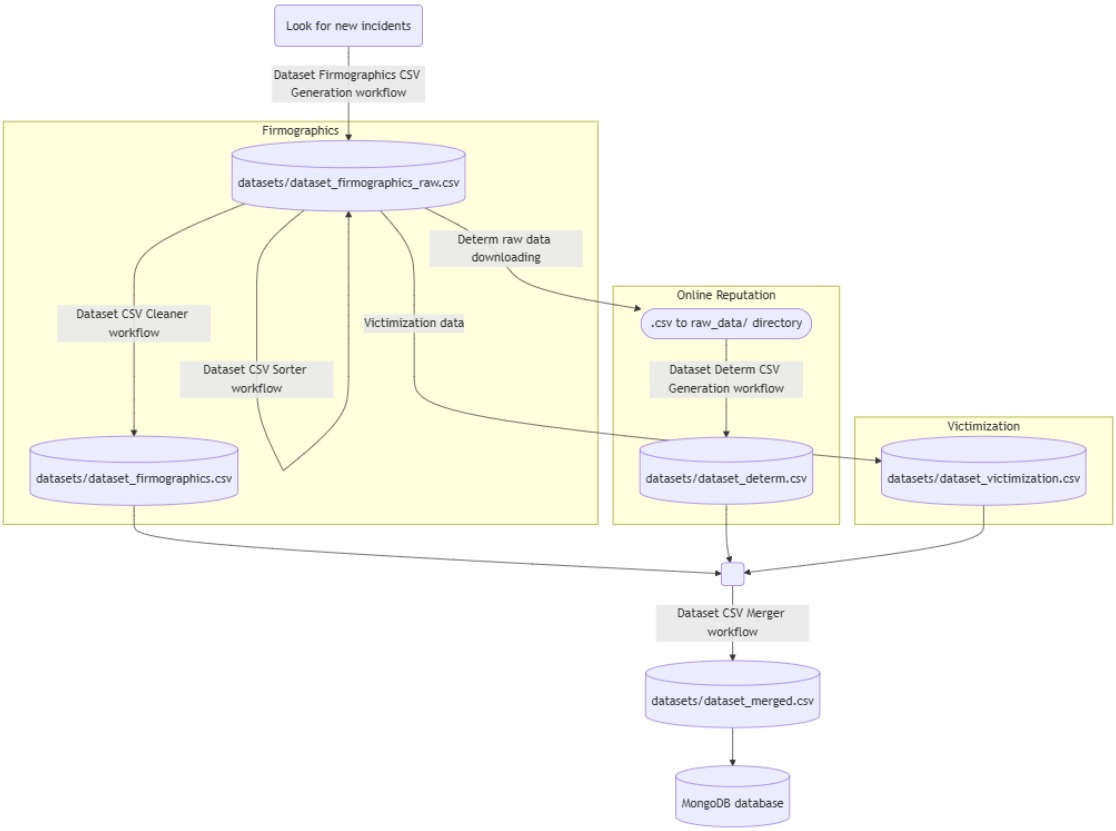

Ilustración 1. Flujo del ETL para atractivo

3.2.1 Flujo de trabajo de extracción de datos

El archivo *dataset_firmographics_action.yml* actúa como orquestador de la etapa de extracción de los datos para atractivo. Este GitHub Workflow se ejecuta manualmente al recibir un nuevo incidente, junto a su fecha y categoría de incidente. Tras instalar las librerías de Python necesarias y configurar las variables de entorno para la autenticación a partir de secretos del Workflow, ejecuta *gen_dataset_firmographics.py*, que se encarga de utilizar las librerías propias *rocketreach* y *companiesmarketcap*, responsables de la extracción de los datos firmográficos en las herramientas homónimas. Como *RocketReach* proporciona —en algunos casos— empresas similares a la buscada, se emplea este campo para definir a los no incidentes. No obstante, se han añadido controles adicionales para asegurar la calidad de este dato, detalladas en el entregable E1.4 Documentación de las técnicas y transformaciones de datos en información para su consumo en modelos de ciber riesgo OT. Por tanto, si alguna comprobación no se cumple o *RocketReach* no conoce empresas similares a la víctima, no se extraen datos de no incidente en dicho flujo.

Una vez obtenido el no incidente, se extraen tanto para la víctima como para esta otra (en caso de existir) los datos de menciones en redes sociales y medios en línea, así como los dispositivos visibles en Internet y los *leaks* de *ransomware* de cada entidad.

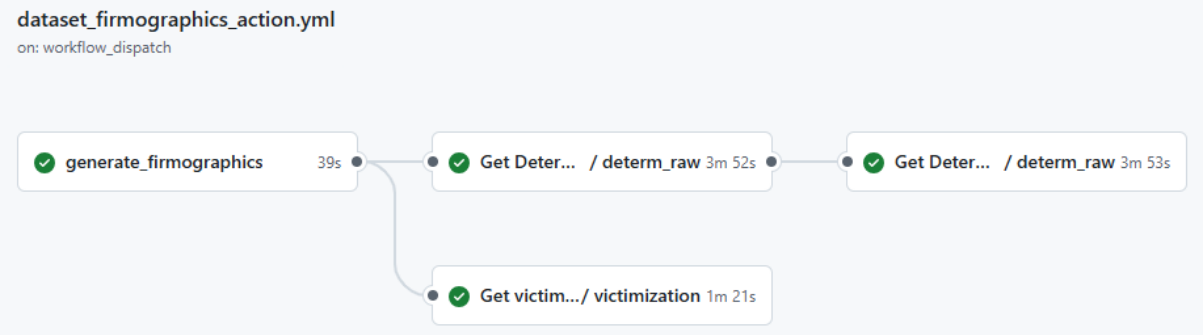

Ilustración 2. GitHub Workflow de extracción de datos

3.3 Flujo de trabajo de ETL de CVE2ATK

Por un lado, se extrae datos de CVE públicos de las bases de datos MITRE CVE y NIST NVD. La base de datos de *Common Vulnerabilities and Exposures* (CVE) asigna un identificador global a cada vulnerabilidad. Utilizamos la *National Vulnerability Database* (NVD) de NIST para obtener información sobre una vulnerabilidad, en particular el ID de CVE, los CWEs relacionados y la descripción. La base de datos de *Common Weaknesses and Exposures* (CWE) se utiliza para obtener los CAPEC asociados. La base de datos de *Common Attack Pattern Enumerations and Classifications* (CAPEC) se emplea obtener las técnicas de MITRE ATT&CK asociadas a un CAPEC. Encadenamos estas bases de datos de la siguiente manera: CVE-NVD/CWE/CAPEC/TTs, con el fin de obtener datos sobre pares de vulnerabilidades y técnicas para nuestro conjunto de datos de entrenamiento del sistema CVE2ATK.

Por otro lado, mediante un sencillo *script* en Python, se obtienen las tácticas y técnicas de manera relacionada entre sí. Para ello, primero se obtiene el listado de tácticas incluidas en cada una de las matrices *Enterprise* e *ICS* accediendo al listado disponible en el sitio web de MITTRE ATT&CK[3, 4]. De dichas páginas se obtiene el identificador de cada táctica, así como su nombre y descripción. Seguidamente, empleando cada identificador, se construye la URL de detalles de cada táctica siguiendo el patrón *f'https://attack.mitre.org/tactics/{tactic_id}/'*, desde el cual se obtienen cada técnica asociada. De nuevo, tanto para técnicas como sub-técnicas se extraen el identificador, nombre y descripción. De esta manera, se obtiene un conjunto de datos que relaciona cada táctica con las técnicas y sub-técnicas asociadas, incluyendo de cada una su identificador en MITRE ATT&CK, nombre y descripción. Para el desarrollo, se han empleado otras utilidades de código abierto desarrolladas por la comunidad como los repositorios de código *mittreattack-python*[5] y *attack-scripts*[6].

3.4 Herramientas y tecnologías empleadas

El desarrollo del ETL se apoya en herramientas de código abierto y tecnologías robustas:

[3] https://attack.mitre.org/tactics/enterprise/
[4] https://attack.mitre.org/tactics/ics/
[5] https://mitreattack-python.readthedocs.io/
[6] https://github.com/mitre-attack/attack-scripts

- **Lenguaje de programación**: Python, aprovechando bibliotecas como Pandas para la transformación de datos, Requests para las conexiones con APIs y Selenium para la navegación automatizada.
- **Gestión de flujos de trabajo**: *GitHub Workflows*, configurado para ejecutar el pipeline automáticamente en base a desencadenantes predefinidos.
- **Bases de datos**: GitHub para almacenamiento inicial de conjuntos de datos parciales y como copia de seguridad, y MongoDB para información semiestructurada y enriquecida para emplear en etapas posteriores como la herramienta de visualización y soporte a la toma de decisiones.
- **Control de versiones**: GitHub para la gestión de código y versiones de los *scripts*.

Estas herramientas se seleccionaron para maximizar la interoperabilidad, la escalabilidad y la facilidad de mantenimiento, así como por las habilidades de ambos equipos y los recursos físicos y económicos disponibles.

3.5 Seguridad y escalabilidad

Dado el carácter crítico de los datos manejados, se incorporaron medidas de seguridad avanzadas:

- **Cifrado en tránsito**: Todos los datos transmitidos entre fuentes, el ETL y las bases de datos utilizan HTTPS/TLS.
- **Control de acceso**: Autenticación basada en roles a los repositorios de código y bases de datos para limitar el acceso a los componentes del ETL.
- **Tolerancia a fallos**: Implementación de reintentos automáticos y recuperación en caso de errores durante la ejecución.

El diseño modular permite integrar nuevas fuentes de datos sin necesidad de cambios significativos en la arquitectura del sistema, garantizando su escalabilidad a largo plazo.

4 DATOS Y CÓDIGO DISPONIBLES

En el enlace deriskGroup / DICyME Project · GitLab se pueden encontrar los siguientes elementos, dentro del directorio E.1.3 Final prototype of the automatic data extraction system:

1. Automations: *Archivos .yml* de extracción de datos y orquestación del proceso.
2. Data: Muestras de los datos extraídos con el sistema.

Para el control de versiones de las dependencias, se ha empleado el microformato *requirements.txt* de Python, en el que se especifican tanto las dependencias como las versiones a emplear. Además, se ha empleado la utilidad *Dependabot*, que detecta versiones de librerías obsoletas y genera los cambios necesarios en requirements.txt para actualizar a las nuevas, entre otras funcionalidades. Esto ayuda a utilizar librerías libres de errores o vulnerabilidades, así como emplear las mismas versiones a lo largo de todo el repositorio de código —en todos los *scripts* y *GitHub Workflows*—.

Algunas de las versiones más relevantes se detallan a continuación:

- *Python 3.11*
- *Beautiful Soup 4.12.3*
- *Numpy 2.1.1*
- *Pandas 2.2.2*
- *Requests 2.32.3*
- *Selenium 4.24.0*

Para poder ejecutar este código sin errores se recomienda usar el mismo entorno que el de desarrollo, arriba señalado.

Cabe destacar que este es un esfuerzo continuo ya que cada semana se visitan las páginas web de ciber incidentes y se actualizan los datos recogidos. En cambio, para la extracción de datos relacionado con el "Atractivo", esta se lleva a cabo una única vez por incidente identificado. Los datos están siendo guardados en una base de datos MongoDB en los servidores de la URJC, además de la replicación en GitHub.

5 CONCLUSIONES Y SIGUIENTES PASOS

En conclusión, este trabajo ha desarrollado con éxito la consolidación de un sistema ETL funcional que integra múltiples fuentes de datos en flujos automatizados y eficientes. Este prototipo final cumple con los objetivos establecidos en el proyecto, proporcionando datos confiables y actualizados para los modelos de ciber riesgo. Además, el éxito en otras actividades del proyecto constata la utilidad del trabajo realizado en esta etapa.

De esta manera, los resultados preliminares indican que el prototipo es capaz de extraer datos relevantes y de calidad, así como valiosos para los modelos de cuantificación del ciber riesgo. Además, el uso de herramientas como *GitHub Workflows* ha permitido optimizar la ejecución y monitorización del flujo de ejecución, reduciendo tiempos de procesamiento y facilitando la identificación y corrección de errores. También es útil para el control de versiones, integrado en *Dependabot*, que asegura consistencia y actualidad en el uso de librerías y dependencias de terceros.