

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA  
DE TELECOMUNICACIÓN  
TRABAJO FIN DE MÁSTER**

**DESARROLLO DE UN ENTORNO PARA  
LA GESTIÓN Y EL TRATAMIENTO  
DINÁMICO DE INTELIGENCIA DE  
AMENAZAS, RIESGOS Y ANOMALÍAS  
BASADO EN ONTOLOGÍAS**

**PAULA GARCÍA FERNÁNDEZ  
2020**



**MÁSTER UNIVERSITARIO EN INGENIERÍA DE  
TELECOMUNICACIÓN**

**TRABAJO FIN DE MÁSTER**

**Título:** Desarrollo de un entorno para la Gestión y el Tratamiento Dinámico de Inteligencia de Amenazas, Riesgos y Anomalías basado en Ontologías  
**Autor:** Dª. Paula García Fernández  
**Tutor:** D. Víctor A. Villagrá  
**Ponente:** D. .....  
**Departamento:** Departamento de Telemática ETSIT UPM

**MIEMBROS DEL TRIBUNAL**

**Presidente:** D. Juan Carlos Dueñas López

**Vocal:** D. Luis Bellido Triana

**Secretario:** D. Óscar Araque Iborra

**Suplente:** D. Sergio Muñoz López

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de: 10

Madrid, a 7 de julio de 2020

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA  
DE TELECOMUNICACIÓN  
TRABAJO FIN DE MÁSTER**

**DESARROLLO DE UN ENTORNO PARA LA  
GESTIÓN Y EL TRATAMIENTO DINÁMICO DE  
INTELIGENCIA DE AMENAZAS, RIESGOS Y  
ANOMALÍAS BASADO EN ONTOLOGÍAS**

**PAULA GARCÍA FERNÁNDEZ  
2020**



## **RESUMEN**

Las organizaciones se exponen actualmente a amenazas y riesgos que cada vez son más difíciles de detectar. La finalidad del análisis de riesgos de toda organización es mantener el sistema en un nivel aceptable de riesgo en el tiempo. Sin embargo, la elaboración de una gestión de riesgos que se adapte al dinamismo de las amenazas o a la complejidad de los sistemas, que ahora procesan grandes cantidades de información, es cada vez más complicada. La aparición de nuevas amenazas APT dificultan el éxito de los procesos de seguridad de una organización. Ya no se trata de detectar amenazas sino de comprender sus TTP, técnicas, tácticas y procedimientos. La información de amenazas debe poder relacionarse en tiempo real con los datos asociados a los riesgos a los que está expuesta una organización.

Si se sigue esta manera de actuar frente a las amenazas, se genera una conciencia situacional más realista y en tiempo real más conforme a la dinámica de los sistemas actuales. Además, existen muchos sistemas de seguridad que ofrecen distintas herramientas para detectar y prevenir amenazas. Dicha información debe ser analizada adecuadamente, ya que de lo contrario no genera beneficios en la batalla contra las amenazas.

Las organizaciones también se han visto involucradas en el avance de la tecnología y el número de sistemas y dispositivos es cada vez mayor. Esta gran cantidad de activos conectados dificulta su protección.

En este contexto surgen las ontologías como sistemas expertos que permiten la representación formal de toda esa información en una estructura común. Las ontologías permiten integrar información de distintos formatos y sintaxis y serán de gran utilidad en entornos heterogéneos.

Este trabajo pretende consolidar una serie de recursos y herramientas en un sistema integrado de datos heterogéneos con el fin de relacionar distintos tipos de información: inteligencia de amenazas, anomalías y datos relacionados con la gestión de riesgos. Para ello, se propone el uso de ontologías y del lenguaje STIX.

El sistema integrará información de fuentes externas que será procesada mediante reglas y motores de inferencia con el fin de detectar amenazas y riesgos. El sistema será capaz de calcular el riesgo dinámicamente y de ofrecer estrategias de actuación frente a los riesgos detectados. Para facilitar la interpretación de los resultados se desarrollará una interfaz gráfica.

El objetivo principal del trabajo es ofrecer una visión global del estado del sistema y de su exposición al riesgo, estableciendo una conciencia situacional en tiempo real.

## **PALABRAS CLAVE**

**Ontologías, SWRL, OWL, Protégé, Inteligencia de Amenazas, STIX, anomalías, OWLAPI**

## SUMMARY

Today's organizations are exposed to a big number of threats and risks which are becoming increasingly difficult to detect. The risk analysis goal of any organization aims to keep the system at an acceptable risk level through time. However, the development of risk management is complicated due to it has to cope with the dynamism of threats and the complexity of the systems. The emerging of new APT threats challenges the success of an organization's security processes. It is no longer a question of detecting threats but of understanding their TTP. Threat information must be able to be correlated in real time with risks data.

This approach leads to more realistic, real-time situational awareness that is more in line with the dynamics of today's systems. In addition, there are many security systems that offer different tools to detect and prevent threats. Such information must be properly analysed, otherwise it will not bring benefits in the battle against threats.

Organizations have also been involved in the progress of technology and in the increasing number of systems and devices. This large number of connected assets complicates their protection.

In this context, ontologies emerge as expert systems that allow formal representation of information in a uniform structure. Ontologies allow the integration of information from different formats and syntax and will be very valuable in heterogeneous environments.

This work aims to consolidate a combination of resources and tools in an integrated system containing heterogeneous data in order to correlate different types of information: threat intelligence, anomalies and risk management data. To this end, the use of ontologies and the STIX language is proposed.

The system will integrate information from external sources that will be processed through rules and inference engines in order to detect threats and risks. The system will be able to calculate the risk dynamically and to offer action strategies against the detected risks. A graphic interface will be developed to facilitate the interpretation of the results.

The main objective of this work is to provide a global vision of the system's state and its exposure to risk, by establishing situational awareness in real time.

## KEYWORDS

**Ontologies, SWRL, OWL, Protégé, Threat Intelligence, STIX, anomalies, OWLAPI**

# ÍNDICE DEL CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN Y OBJETIVOS .....</b>	<b>1</b>
1.1	Introducción .....	1
1.2	Objetivos.....	2
1.3	Estructura de la memoria .....	2
<b>2</b>	<b>CIBERSEGURIDAD Y GESTIÓN DINÁMICA DE RIESGOS.....</b>	<b>4</b>
2.1	Seguridad en Redes de Telecomunicación.....	4
2.2	Análisis y Gestión de Riesgos.....	5
2.2.1	Tecnologías para el análisis y gestión de riesgos.....	6
2.3	Inteligencia de Amenazas .....	7
2.3.1	Tecnologías para la Inteligencia de Amenazas .....	8
<b>3</b>	<b>ONTOLOGÍAS: MODELOS FORMALES DE REPRESENTACIÓN DE LA INFORMACIÓN .....</b>	<b>12</b>
3.1	Lenguajes de Definición de Ontologías .....	14
3.1.1	Resource Description Framework (RDF)/RDF-Schema (RDFS) .....	14
3.1.2	Ontology Interchange Language (OIL) .....	15
3.1.3	DAML + OIL .....	15
3.1.4	Ontology Web Language (OWL) .....	15
3.1.5	OWL 2.....	17
3.2	Lenguajes de definición de Reglas de comportamiento .....	18
3.2.1	JESS .....	18
3.2.2	RuleML .....	18
3.2.3	R2ML .....	19
3.2.4	SWRL .....	19
3.3	Tecnologías para el Manejo y Gestión de Ontologías Y Reglas de Comportamiento .....	21
3.3.1	Editor Protégé .....	21
3.3.2	OWL API .....	21
3.3.3	Jena .....	22
3.3.4	Razonadores semánticos .....	22
<b>4</b>	<b>DISEÑO GENERAL DEL SISTEMA .....</b>	<b>26</b>
4.1	Ámbito del proyecto .....	27
4.2	Requisitos .....	28
4.3	Casos de Uso.....	29
4.4	Arquitectura Del Sistema.....	34
<b>5</b>	<b>DISEÑO DE LA ONTOLOGÍA .....</b>	<b>38</b>
5.1	Definición de la Ontología .....	38
5.1.1	Clases de la ontología .....	39
5.2	Métricas de seguridad .....	47
5.2.1	Reglas de Comportamiento .....	47

<b>6</b>	<b>DESARROLLO DEL SISTEMA .....</b>	<b>56</b>
6.1	Implementación en Protégé .....	56
6.2	Implementación de <i>Parsers</i> para la introducción de datos en la Ontología.....	62
6.2.1	Parser para las fuentes de datos de anomalías .....	62
6.2.2	Parser para las fuentes de datos de <i>Threat Intelligence</i> .....	64
6.2.3	Parser para los activos procedentes de la herramienta PILAR.....	65
6.3	Cálculo Dinámico del Riesgo.....	66
6.4	Razonamiento Semántico.....	68
<b>7</b>	<b>VALIDACIÓN.....</b>	<b>69</b>
7.1	Integración de información de activos .....	69
7.2	Integración de información de anomalías procedente de fuentes externas.....	72
7.3	Integración de información procedente de fuentes de <i>Threat Intelligence</i> .....	75
7.4	Creación de amenazas y riesgos.....	77
7.5	Cálculo del Riesgo .....	82
7.6	Interfaz de Visualización.....	83
7.7	Caso Práctico .....	86
<b>8</b>	<b>CONCLUSIONES .....</b>	<b>91</b>
8.1	Conclusiones .....	91
8.2	Líneas futuras .....	92
<b>9</b>	<b>REFERENCIAS.....</b>	<b>93</b>
<b>ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES .....</b>		<b>95</b>
A.1	Introducción .....	95
A.2	Descripción de impactos relevantes relacionados con el proyecto.....	95
A.3	Conclusiones.....	97
<b>ANEXO B: PRESUPUESTO ECONÓMICO .....</b>		<b>98</b>

# ÍNDICE DE FIGURAS

FIGURA 1. PROCESO DE ANÁLISIS DE RIESGOS [4] .....	6
FIGURA 2. REPRESENTACIÓN DE LA ARQUITECTURA STIX [4] .....	10
FIGURA 3. SISTEMA DE ONTOLOGÍAS.....	13
FIGURA 4. ARQUITECTURA GLOBAL DEL PROYECTO EN EL QUE SE ENMARCA EL TFM .....	27
FIGURA 5. DIAGRAMA DE SECUENCIA DEL CASO DE USO CU01 .....	30
FIGURA 6. DIAGRAMA DE SECUENCIA DEL CASO DE USO CU02 .....	31
FIGURA 7. DIAGRAMA DE SECUENCIA DEL CASO DE USO CU03 .....	32
FIGURA 8. DIAGRAMA DE SECUENCIA DEL CASO DE USO CU04 .....	33
FIGURA 9. DIAGRAMA DE SECUENCIA DEL CASO DE USO CU05 .....	34
FIGURA 10. ARQUITECTURA GENERAL DEL SISTEMA.....	35
FIGURA 11. RELACIONES ENTRE LAS DIFERENTES ONTOLOGÍAS.....	39
FIGURA 12. DEFINICIÓN DE UNA ONTOLOGÍA EN EL EDITOR PROTÉGÉ.....	57
FIGURA 13. CLASES DE UNA ONTOLOGÍA DEFINIDAS EN EL EDITOR PROTÉGÉ.....	57
FIGURA 14. DATATYPE PROPERTIES DE UNA ONTOLOGÍA DEFINIDAS EN EL EDITOR PROTÉGÉ.....	58
FIGURA 15. OBJECT PROPERTIES DE UNA ONTOLOGÍA DEFINIDAS EN EL EDITOR PROTÉGÉ.....	58
FIGURA 16. DATATYPES DE UNA ONTOLOGÍA DEFINIDOS EN EL EDITOR PROTÉGÉ .....	59
FIGURA 17. INDIVIDUALS O INSTANCIAS DE UNA ONTOLOGÍA DEFINIDOS EN EL EDITOR PROTÉGÉ .....	59
FIGURA 18. REPRESENTACIÓN DE LAS CLASES DE LA ONTOLOGÍA ONA MEDIANTE ONTOGRAPH .....	60
FIGURA 19. CLASES DE LA ONTOLOGÍA DRM .....	61
FIGURA 20. CLASES DE LA ONTOLOGÍA CTI .....	61
FIGURA 21. DIAGRAMA UML PARA LA CLASE ANOMALIES.JAVA .....	63
FIGURA 22. DIAGRAMA UML PARA LA CLASE STIX.JAVA.....	65
FIGURA 23. DIAGRAMA UML PARA LA CLASE DRM.JAVA.....	66
FIGURA 24. CREACIÓN DE UNA INSTANCIA DE RISK SCOPE .....	70
FIGURA 25. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN DE ACTIVOS DE PILAR .....	70
FIGURA 26. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN DE VALORACIÓN DE ACTIVOS .....	71
FIGURA 27. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA GENERACIÓN DE VALORACIONES DE ACTIVOS DERIVADAS DEL ACTIVO ESENCIAL.....	71
FIGURA 28. ASOCIACIÓN ENTRE ACTIVOS Y VALORACIÓN DE ACTIVOS .....	72
FIGURA 29. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN SOBRE UNA ANOMALÍA DE TIPO WIFI EN EL SISTEMA.....	74
FIGURA 30. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN SOBRE UNA ANOMALÍA DE TIPO CIBERSEGURIDAD EN EL SISTEMA.....	75
FIGURA 31. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN STIX SOBRE UNA VULNERABILIDAD ..	76
FIGURA 32. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA INTEGRACIÓN DE INFORMACIÓN STIX SOBRE UNA CAMPAÑA.....	77
FIGURA 33. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA DEFINICIÓN DE REGLAS SWRL SOBRE LA CREACIÓN DE RIESGOS DE CUMPLIMIENTO DE PROTECCIÓN DE DATOS.....	77
FIGURA 34. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UNA AMENAZA A PARTIR DE LOS ACTIVOS.....	78
FIGURA 35. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UN RIESGO DE TIPO DATA PROTECTION COMPLIANCE ..	78
FIGURA 36. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA DEFINICIÓN DE REGLAS SWRL PARA ANOMALÍAS.....	79
FIGURA 37. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UNA AMENAZA A PARTIR DE UNA ANOMALÍA. ....	79
FIGURA 38. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA DEFINICIÓN DE REGLAS SWRL PARA INFORMACIÓN STIX .....	80
FIGURA 39. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UNA AMENAZA A PARTIR DE INFORMACIÓN STIX .....	80
FIGURA 40. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UN RIESGO DE TIPO Denial of Service .....	81
FIGURA 41. PRUEBA DE VALIDACIÓN CORRESPONDIENTE A LA CREACIÓN DE UN RIESGO POTENCIAL.....	81
FIGURA 42. INSTANCIA DE TIPO SALVAGUARDA.....	82
FIGURA 43. HISTORIAL DE DATOS RESULTANTES DEL CÁLCULO DEL RIESGO .....	83
FIGURA 44. INTERFAZ DE VISUALIZACIÓN: RIESGOS TOTALES FINALES POTENCIALES Y RESIDUALES.....	84
FIGURA 45. INTERFAZ DE VISUALIZACIÓN: RIESGO INSTANTÁNEO TOTAL POTENCIAL Y RESIDUAL .....	85
FIGURA 46. INTERFAZ DE VISUALIZACIÓN: RIESGO POTENCIAL Y RESIDUAL POR TIPO DE RIESGO .....	85
FIGURA 47. INTERFAZ DE VISUALIZACIÓN: MAPA DE RIESGOS .....	86
FIGURA 48. INTERFAZ DE VISUALIZACIÓN: SOPORTE A LA TOMA DE DECISIONES .....	86
FIGURA 49. ESCENARIO DEL EJEMPLO PRÁCTICO .....	87
FIGURA 50. FLUJO DE LA EJECUCIÓN DEL EJEMPLO PRÁCTICO .....	88
FIGURA 51. RESULTADO DEL EJEMPLO PRÁCTICO.....	88

FIGURA 52. RIESGOS RESULTANTES DEL EJEMPLO PRÁCTICO EN EL SISTEMA DE VISUALIZACIÓN .....	89
FIGURA 53. SOPORTE A LA TOMA DE DECISIONES DEL EJEMPLO PRÁCTICO .....	89

# ÍNDICE DE TABLAS

TABLA 1. COMPARACIÓN DE STIX CON OTROS ESTÁNDARES.....	11
TABLA 2. COMPARACIÓN DE LOS LENGUAJES DE DEFINICIÓN DE ONTOLOGÍAS .....	18
TABLA 3. COMPARACIÓN DE LOS LENGUAJES DE DEFINICIÓN DE REGLAS.....	20
TABLA 4. COMPARACIÓN DE LOS RAZONADORES SEMÁNTICOS. ....	25
TABLA 5. REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.....	28
TABLA 6. REQUISITO DE ENTRADA DEL SISTEMA .....	28
TABLA 7. REQUISITOS DE SALIDA DEL SISTEMA .....	29
TABLA 8. CASO DE USO CU01: DETECCIÓN Y ACTUALIZACIÓN DE ANOMALÍAS .....	29
TABLA 9. CASO DE USO CU02: DETECCIÓN Y ACTUALIZACIÓN DE INFORMACIÓN DE <i>THREAT INTELLIGENCE</i> .....	30
TABLA 10. CASO DE USO CU03: INTRODUCCIÓN DE DATOS DE ACTIVOS .....	31
TABLA 11. CASO DE USO CU04: RAZONAMIENTO SEMÁNTICO DE LA ONTOLOGÍA .....	33
TABLA 12. CASO DE USO CU05: VISUALIZACIÓN DE RESULTADOS .....	33
TABLA 13. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANOMALÍA DETECTADA .....	40
TABLA 14. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANOMALÍA WIFI .....	42
TABLA 15. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANOMALÍA BLUETOOTH.....	43
TABLA 16. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANOMALÍA CIBERSEGURIDAD.....	44
TABLA 17. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANONMALÍA RADIOFRECUENCIA .....	45
TABLA 18. DEFINICIÓN DE LA ONTOLOGÍA. CLASE ANOMALÍA REDES MÓVILES .....	46
TABLA 19. DEFINICIÓN DE LA ONTOLOGÍA. CLASE EVENTO.....	46
TABLA 20. REGLAS DE ANOMALÍAS: ANOMALÍA WIFI .....	49
TABLA 21. REGLAS DE ANOMALÍAS: ANOMALÍA BLUETOOTH .....	49
TABLA 22. REGLAS DE ANOMALÍAS: ANOMALÍA RADIOFRECUENCIA .....	50
TABLA 23. REGLAS DE ANOMALÍAS: ANOMALÍA REDES MÓVILES.....	50
TABLA 24. REGLAS DE ANOMALÍAS: ANOMALÍA CIBERSEGURIDAD.....	51
TABLA 25. REGLAS DE INVENTARIO DE AMENAZAS: DELIBERATED INFORMATION LEAK THREAT .....	51
TABLA 26. REGLAS DE INVENTARIO DE RIESGOS: DATA PROTECTION RISK.....	52
TABLA 27. REGLAS DE INVENTARIO DE RIESGOS: DELIBERATED UNAUTHORIZED ACCESS RISK.....	52
TABLA 28. REGLAS DE INVENTARIO DE RIESGOS: DENIAL OF SERVICE RISK .....	52
TABLA 29. REGLAS DE INVENTARIO DE RIESGOS: DELIBERATED INFORMATION LEAK RISK .....	53
TABLA 30. REGLAS DE EVALUACIÓN DE RIESGOS: RESIDUAL RISK FOR DATA PROTECTION COMPLIANCE RISK .....	54
TABLA 31. REGLAS DE EVALUACIÓN DE RIESGOS: RESIDUAL RISK FOR DENIAL OF SERVICE RISK .....	54
TABLA 32. REGLAS DE EVALUACIÓN DE RIESGOS: RESIDUAL RISK FOR DELIBERATED UNAUTHORIZED ACCESS RISK.....	55
TABLA 33. REGLAS DE EVALUACIÓN DE RIESGOS: RESIDUAL RISK FOR DELIBERATED INFORMATION LEAK RISK.....	55
TABLA 34. COSTES DE RECURSOS HUMANOS .....	98
TABLA 35. COSTES DE RECURSOS MATERIALES .....	99
TABLA 36. COSTES DE MATERIAL FUNGIBLE .....	99
TABLA 37. COSTES GENERALES Y BENEFICIO INDUSTRIAL .....	99
TABLA 38. PRESUPUESTO TOTAL ANTES Y DESPUÉS DE IMPUESTOS.....	100



# 1 INTRODUCCIÓN Y OBJETIVOS

---

## 1.1 INTRODUCCIÓN

La tecnología en la actualidad avanza a un ritmo exponencial. Cada vez son más las organizaciones que incluyen nuevas funcionalidades a sus sistemas y redes de telecomunicaciones. Esto ha sido posible gracias en gran parte al Big Data y Cloud Computing. Millones de datos son procesados aumentando la complejidad de los sistemas de procesamiento. Sin embargo, esta complejidad incrementa la probabilidad de que ocurran nuevos incidentes y errores no intencionados, los cuales se deben tener en cuenta en los procesos de análisis de riesgos de las organizaciones para poder controlarlos y anticiparse a ellos.

Los sistemas de información que componen las organizaciones han ido incrementándose a lo largo de los años y cada vez es más evidente la dependencia entre el número de sistemas y la seguridad. A medida que aumenta el número de elementos de una organización, las medidas de seguridad se vuelven más complejas. La tarea de análisis y gestión de riesgos se vuelve complicada y compleja con la aparición del Internet de las Cosas (IoT). El creciente número de dispositivos conectados pone a disposición de los atacantes la posibilidad de explotar nuevas vulnerabilidades, y conlleva la aparición de nuevas amenazas y riesgos. Además, la ciberseguridad se convierte en un atributo importante para los clientes. Como se describe en el siguiente estudio [1] es importante la facilidad de uso de un producto IoT para su éxito en el mercado. Ahora bien, si la facilidad de uso conlleva al aumento del riesgo, el atractivo de un producto se ve reducido. A la hora de adquirir un producto, la seguridad percibida por los clientes se convierte en un elemento fundamental a tener en cuenta por los proveedores de los nuevos productos IoT.

La evolución de la conectividad conlleva al avance de las amenazas, que cada vez son más sofisticadas y difíciles de detectar. La aparición de las amenazas persistentes avanzadas (APT, *Advanced Persistent Threat*) ha producido una necesidad de mejora de los sistemas de detección, prevención y respuesta a incidentes por parte de las empresas. En [2] el CSIRT-CV e INCIBE elaboran un informe sobre este tipo de ciberamenazas y sus consecuencias, de las cuales ninguna empresa está exenta de peligro, como muestra el estudio de IT Digital Security donde se recogen algunas de las APT ocurridas en los últimos años [3]. Estas ciberamenazas cambian dinámicamente lo que conlleva a que los actuales sistemas de seguridad se vean obsoletos y que los IoCs sean poco eficientes frente a las APT.

Es evidente la necesidad de un cambio en las empresas sobre la manera de evaluar el riesgo al que están expuestas. Existen muchas guías, referencias y métodos para la implantación de medidas de seguridad en una organización, como las normas ISO 27005 e ISO 31000, en las que se basa la metodología MAGERIT [4]. Sin embargo, los métodos actuales son procesos iterativos con períodos largos que tienen en cuenta solo una parte de la organización. Estos marcos antiguos no están a la altura del dinamismo actual. Cuando ocurre un incidente no se relaciona con los procesos de evaluación y gestión de riesgos, lo que produce limitaciones en la lucha contra las amenazas. Ya que no se trata solo de detectarlas sino de comprender su comportamiento, sus Técnicas, Tácticas y procedimientos (TTP), lo cual requiere tiempo e inversión monetaria.

Por otro lado, el dinamismo de las amenazas hace imprescindible la información que se tiene sobre ellas. La información de inteligencia de amenazas (CTI) cada vez es más importante [5] [6]. El 80% de las empresas considera que la CTI ha mejorado su capacidad de detección de amenazas y un 40% piensa que la mejor forma de responder frente a una amenaza es el uso de IoCs, mientras un 20% da más importancia a la comprensión de las TTP del adversario. En los últimos años han surgido nuevos estándares que van más allá de los IoCs y que cuentan con gran apoyo por parte de organizaciones destacadas en el ámbito de la ciberseguridad, como el estándar STIX. Esto se debe a que cuanta más información se tenga sobre amenazas, más fiable es un sistema y más probable será que el sistema pueda responder de manera proactiva. Es aquí donde también surge una llamada a las organizaciones a compartir esta información entre ellas. Esto se debe a que organizaciones similares podrán ser víctimas de las mismas amenazas.

Para ello, se necesitan además sistemas que lleven a cabo procesos complejos que engloban una gran cantidad de información. Estos procesos deberán ser lo más automáticos posibles para poder dar una respuesta en tiempo real o en casi tiempo real. Los procesos de seguridad evolucionan y a medida que se descubre información de amenazas nuevas se retroalimentan los procesos y los sistemas y se mejora la detección, la gestión y la respuesta frente a riesgos, amenazas y anomalías. En este contexto, surgen las ontologías que permiten representar toda esa información en estructuras unificadas para ser procesada posteriormente.

Este trabajo se enmarca en un proyecto más amplio que pretende monitorizar múltiples fuentes de información tanto estáticas como dinámicas, procesar los datos recolectados mediante técnicas de aprendizaje automático y sistemas expertos de ontologías, detectar y predecir patrones avanzados de ataques y calcular el riesgo de la exposición a los mismos. El contexto de este proyecto ha marcado en gran medida el diseño del sistema del presente trabajo.

## 1.2 OBJETIVOS

En primer lugar, se necesita realizar una serie de estudios y análisis previos a la implementación del sistema para después elegir las tecnologías más adecuadas conforme a la implementación del sistema propuesto:

- Estudiar el estado actual de la seguridad en las telecomunicaciones, en concreto el entorno que enmarca la gestión de riesgos y la inteligencia de amenazas, así como los lenguajes y herramientas que se enmarcan en dicho contexto.
- Analizar los sistemas de ontologías y estudiar las métricas de seguridad existentes en la actualidad.
- Proponer un sistema de ontologías que se adapte a los requisitos expuestos y seleccionar las tecnologías más adecuadas, tales como lenguajes para la definición de las ontologías, las métricas de seguridad y herramientas de programación. Estudiar dicha tecnología para poder llevar a cabo la implementación de un sistema de ontologías que sea capaz de integrar información de fuentes externas.
- Desarrollar el sistema completo que comprende el sistema de ontologías, el sistema de cálculo del riesgo, el sistema de soporte a la toma de decisiones y el sistema de visualización.
- Validar la funcionalidad del sistema mediante la ejecución de pruebas.

## 1.3 ESTRUCTURA DE LA MEMORIA

La memoria se estructura en seis capítulos:

### **Capítulo 2: Ciberseguridad y Gestión Dinámica de Riesgos**

En este capítulo se abordarán conceptos y términos sobre la seguridad de la información necesarios para la comprensión del trabajo. Por un lado, se realizará un análisis de las tecnologías destinadas a la gestión de riesgos, prestando especial atención a las herramientas para la gestión tanto dinámica como estática de los riesgos. Por otro lado, se analizarán las implicaciones de la inteligencia de amenazas en la detección temprana de amenazas y los lenguajes utilizados para compartir dicha

información. Se realizará un cuadro comparativo entre estos lenguajes que después será utilizado para justificar la elección del lenguaje más adecuado para el trabajo.

### **Capítulo 3: Ontologías: Modelos Formales de Representación de la Información**

En este capítulo se realizará un análisis del estado actual de los sistemas de ontologías. Este análisis incluye un estudio de los lenguajes formales existentes para la definición de ontologías, de las métricas de seguridad más utilizadas y de las tecnologías existentes para la gestión de las ontologías. Cada parte concluirá con una comparación entre conceptos similares que será utilizada para justificar la elección de unos sobre otros en el desarrollo del trabajo.

### **Capítulo 4: Diseño del sistema.**

En este capítulo se justificarán cada una de las partes utilizadas. En primer lugar, se detallará el ámbito del proyecto en el que se enmarca el trabajo. Por otro lado, se determinarán cada uno de los requisitos que debe cumplir el sistema. Los requisitos se detallarán con los casos de uso y los diagramas de secuencia derivados. Por último, se presentará la arquitectura general del sistema, la cual se subdivide en subsistemas, cada uno de los cuales tiene una misión determinada. Este conjunto de subsistemas permitirá cumplir con los objetivos principales del sistema.

### **Capítulo 5: Diseño de la ontología.**

En este capítulo se especificará el diseño de la ontología. La ontología es la base de este proyecto por lo que se dedica el capítulo al completo para la definición formal de la ontología, las clases que la componen, así como las métricas de seguridad implementadas, que incluyen la definición de las reglas de comportamiento.

### **Capítulo 6: Desarrollo del Sistema.**

En este capítulo se presentará el desarrollo de cada una de las partes del sistema. Este capítulo se detallará siguiendo el orden en el que se ha procedido para desarrollar del sistema. En primer lugar, se detallará la implementación de la ontología final propuesta. A continuación, se detallará la implementación de los distintos *parsers* para la integración de los datos procedentes de fuentes externas en la ontología definida. Por otro lado, se explicará el subsistema dedicado al cálculo del riesgo. Por último, se detallarán las implicaciones del razonador semántico para la inferencia de nuevo conocimiento.

### **Capítulo 7: Validación.**

En este capítulo se someterá al sistema a una serie de pruebas para verificar que cumple con los objetivos. Se presentará además un caso práctico demostrativo de la funcionalidad del sistema. De esta manera, se validará la propuesta de sistema del trabajo.

### **Capítulo 8: Conclusiones.**

Este último capítulo resaltará los puntos más importantes del trabajo, así como las conclusiones que se extraen de la realización del proyecto. Finalmente, se proponen líneas futuras de investigación y continuación del proyecto.

## 2 CIBERSEGURIDAD Y GESTIÓN DINÁMICA DE RIESGOS

### 2.1 SEGURIDAD EN REDES DE TELECOMUNICACIÓN

La seguridad de la información es el conjunto de medidas preventivas y reactivas que permiten proteger un entorno y controlan que los recursos de un sistema de información de una organización sean utilizados según las políticas establecidas por la misma. Puesto que día a día surgen nuevas amenazas y cada vez más complejas, la situación de riesgo cero en una empresa es inalcanzable. Por el mero hecho de existir la empresa ya tiene un riesgo. Un sistema de información se considera seguro en la medida que preserve la confidencialidad, integridad, autenticidad y disponibilidad de sus datos.

Para afrontar la estrategia de seguridad de la información se llevan a cabo planificaciones de seguridad. Estas planificaciones pretenden proteger los sistemas de información de una organización de manera integral y existen normas y recomendaciones que permiten implantar políticas de seguridad para luchar contra las nuevas amenazas.

El avance de las amenazas produce la aparición de nuevas amenazas más complejas que ya no buscan una repercusión mediática sino obtener un beneficio monetario de su ataque. Estas son las denominadas APT (*Advanced Persistent Threat*). Las APT se caracterizan por tener un objetivo identificado y concreto y suponen ataques meditados y sofisticados. Estos ataques pretenden tener presencia durante un periodo de tiempo prolongado en los sistemas de la víctima, lo que puede conllevar consecuencias potencialmente desastrosas.

Teniendo en cuenta que los efectos de estas amenazas pueden ser igual de destructivos para organizaciones similares, cada vez son más las organizaciones que buscan compartir información de inteligencia de amenazas. La *Cyberthreat Intelligence (CTI)* o Inteligencia de Ciberamenazas se encarga de realizar un análisis exhaustivo y una comprensión y caracterización profunda de las técnicas y herramientas utilizadas por los atacantes contra una organización. Son muchos los beneficios de utilizar la CTI en una organización. En [7] se analiza la incorporación de CTI en las operaciones de ciberseguridad a lo largo de las diferentes fases de un ataque (*kill chain phases*) y se concluye que, en los ataques más complejos, el uso de CTI era relevante para reducir el éxito del atacante. La CTI tiene un papel muy importante en la detección de APT y disminuye la probabilidad de que un ataque tenga éxito.

La inteligencia de amenazas provee a las operaciones de ciberseguridad de un aspecto proactivo que permite a las organizaciones anticiparse a sus atacantes y establecer medidas de seguridad más severas y actualizadas. Además, esta inteligencia contribuye a que una organización sea consciente de su exposición a las ciberamenazas y las detecte lo antes posible para llevar a cabo una buena gestión del riesgo.

Ahora bien, si el análisis de riesgos y su gestión no tienen en cuenta el aspecto dinámico de las amenazas, la organización se aleja de la capacidad de percibir una conciencia situacional en tiempo real. Es necesario tener en cuenta la naturaleza dinámica de las amenazas y de los incidentes para poder dar una respuesta en tiempo real o en quasi tiempo real.

Por otro lado, los nuevos avances en el procesamiento automático con *machine learning* también ofrecen apoyo en la detección de anomalías en los sistemas de telecomunicaciones. Además, esta automatización de los procesos permite realizar un análisis de riesgos dinámico ya que se puede realizar de manera mucho más rápida y en tiempo real o casi tiempo real. También en este proceso automático se tienen cuenta un mayor número de datos lo que favorece que dicho análisis sea más fiable. Estos sistemas de procesamiento son entrenados mediante información procedente de otros sistemas, como IDS o sistemas de sensores, para revelar la existencia de comportamientos anómalos que pudieren ser indicadores de posibles amenazas.

Dado que son muchos los sistemas que ofrecen escaneo de vulnerabilidades, gestión de activos, análisis y gestión de riesgos y plataformas de compartición de información de amenazas, surge la necesidad de fusionar toda esa información de seguridad en sistemas estructurados. Comienzan a aparecer sistemas expertos que representan formalmente toda esa información en estructuras para llevar a cabo tareas en el ámbito técnico, operacional y estratégico. Estos sistemas son los denominados sistemas de ontologías, que son la base de este proyecto.

## 2.2 ANÁLISIS Y GESTIÓN DE RIESGOS

El uso de las tecnologías de la información implica el aumento de los riesgos en los sistemas de información. Por ello, la gestión de riesgos ha cobrado gran importancia en el proceso empresarial de cualquier organización.

Un análisis de riesgos es un proceso que se encarga de identificar y evaluar las situaciones e incidentes que pudieran generar un impacto negativo sobre la organización. La gestión de riesgos apoya el buen gobierno de la organización. Contribuye en la toma de decisiones y en la implantación de medidas preventivas y correctivas para garantizar la seguridad de la información.

Sin embargo, para que la seguridad de la información sea establecida adecuadamente en una organización, se necesita llevar a cabo un proceso de planificación de seguridad. En este proceso se indica qué se quiere proteger, cuánto cuesta y cómo se debe realizar, lo que implica la ejecución de un análisis de riesgos.

Existen muchas guías y metodologías en España que recomiendan maneras de realizar un análisis de riesgos adecuado para objetivar cuál es el nivel de riesgo al se está expuesto. Sin embargo, muchas de ellas se caracterizan por ser análisis estáticos. La metodología MAGERIT es un ejemplo. Con esta metodología se consigue solo analizar un escenario estático donde no se tienen en cuenta posibles cambios en el sistema como la ocurrencia de incidentes. MAGERIT permite realizar un cálculo del riesgo de una organización. Sin embargo, la necesidad de llevar a cabo un análisis y un cálculo dinámico del riesgo es indudable ya que es esta la manera para poder realizar un análisis eficiente de la información de inteligencia de amenazas compartida y una gestión del riesgo en tiempo real.

El punto de partida de un análisis de riesgos es el proceso de identificación de los activos que serán los recursos fundamentales para que la organización cumpla con sus objetivos. Para ello, existen herramientas que automatizan la identificación y valoración de activos y a partir de ella deducir las amenazas y riesgos a los que están expuestos dichos activos.

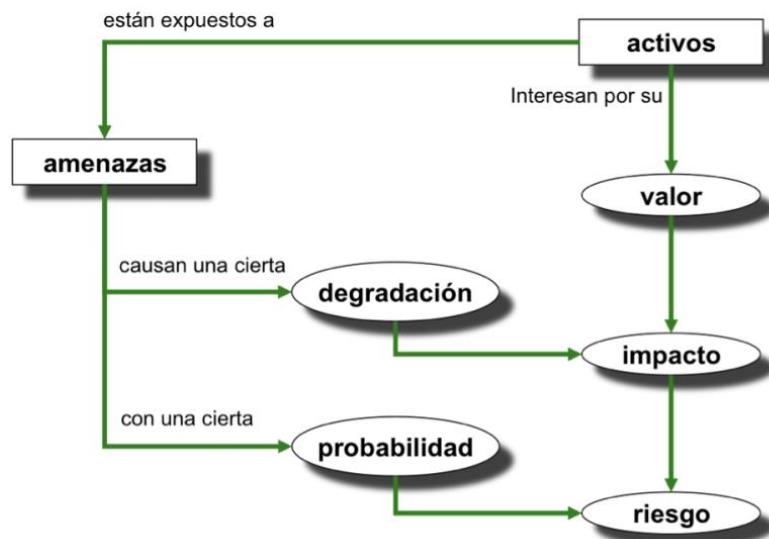


Figura 1. Proceso de análisis de riesgos [4]

Si bien las operaciones de ciberseguridad son complejas a medida que va evolucionando la tecnología, hay una gran cantidad de herramientas de ciberseguridad que facilitan la prevención, detección y respuesta a incidentes de seguridad, como herramientas de gestión de vulnerabilidades, generación de modelos e informes, etc. Estas realizan procesos de análisis iterativos sobre los activos con el fin de detectar vulnerabilidades que puedan ser explotadas por agentes maliciosos causando un impacto negativo sobre la organización.

### 2.2.1 TECNOLOGÍAS PARA EL ANÁLISIS Y GESTIÓN DE RIESGOS

Sin embargo, algunos de estos procesos se caracterizan por ser dinámicos y otros por ser estáticos lo que limita la gestión de riesgos. A continuación, se analizan algunas de estas herramientas.

#### 2.2.1.1 GESTIÓN DINÁMICA DE RIESGOS

La gestión de vulnerabilidades en las organizaciones es una tarea fundamental para prevenir ataques. Se realizan auditorías y test de penetración en busca de nuevas vulnerabilidades en los sistemas para después ser gestionadas en los centros de operaciones de ciberseguridad, SOC. En ellos se analizan las amenazas derivadas de estas vulnerabilidades y se decide el tratamiento a llevar a cabo.

Existen herramientas para la gestión de vulnerabilidades como Nessus y OpenVAS. Nessus [8] escanea las vulnerabilidades existentes en los activos de una organización que puedan ponerles en compromiso. Nessus es un programa de escaneo de vulnerabilidades compuesto por un demonio y un cliente. El demonio realiza el escaneo y el cliente informa de los avances en dicho escaneo. Nessus reduce el tiempo de evaluación de vulnerabilidades para priorizar sobre la reparación de los problemas. Tiene una amplia cobertura de vulnerabilidades y cuenta con un gran apoyo por parte de las organizaciones.

OpenVAS [9] consiste en un software que dispone de herramientas y servicios especializados para realizar escaneos y gestión de vulnerabilidades de sistemas informáticos. OpenVAS se ofrece de forma gratuita con Kali Linux. Su funcionamiento es similar a Nessus, dispone de dos servicios: servidor y cliente. El primero realiza el escaneo y el segundo es utilizado por el usuario para exponer los resultados de dichos escaneos.

La diferencia entre ambos radica en que OpenVAS es una rama de la versión 2 de Nessus que se ofrece como código abierto frente a Nessus que es una herramienta de código cerrado. Si bien el número de vulnerabilidades descubiertas con la herramienta Nessus es mayor, usar una herramienta u otra depende de las necesidades. Sin embargo, son muchos los organismos como el NIST que animan al uso conjunto de varias herramientas de escaneo de vulnerabilidades ya que se complementan en el descubrimiento de vulnerabilidades realizando un escaneo eficiente y más exhaustivo.

#### 2.2.1.2 GESTIÓN ESTÁTICA DE RIESGOS

Debido al aumento considerable de dispositivos conectados se incrementa también la posibilidad de explotar nuevas vulnerabilidades o generar riesgos que se propaguen entre los dispositivos conectados. Por ello, es fundamental llevar a cabo un inventario de activos que permita llevar la cuenta de los activos de la organización y con ello controlar los riesgos que les afecten.

PILAR (Procedimiento Informático Lógico de Análisis de Riesgos) es una herramienta de análisis de riesgos que implementa la metodología MAGERIT y pertenece al conjunto de herramientas que ofrece el CCN [10]. PILAR ofrece un análisis de riesgos cuantitativo y cualitativo y permite generar un inventario de activos para la identificación de posibles amenazas y riesgos.

PILAR genera un marco para la valoración de los riesgos detectados sobre los activos de una organización. Supone una base de datos de activos y de amenazas y riesgos a los que se exponen. PILAR también genera informes a partir de los datos introducidos: informes de impacto, informes de riesgos, evaluación de salvaguardas, informe de insuficiencias o salvaguardas, etc. Además, permite exportar en formatos CSV o XML algunos de los datos generados, como los activos, el valor acumulado de activos, las amenazas, etc.

### 2.3 INTELIGENCIA DE AMENAZAS

Como ya se ha comentado en el apartado anterior, la *Threat Intelligence* es una parte de la ciberseguridad que se centra en el análisis y la recopilación de información sobre ciberataques, así como en la comprensión y caracterización de sus técnicas, tácticas y procedimientos (TTP). Las TTP permiten caracterizar la forma en la que operan los adversarios en el ciberespacio, estableciendo qué es lo que hacen y cómo lo hacen.

Las TTP se encuentran en el eslabón más alto de la llamada pirámide del dolor. El concepto *pirámide de dolor* refleja la relación entre los distintos indicadores utilizados para detectar incidentes y el daño potencial que supone utilizar dichos indicadores contra un atacante. Es decir, si los indicadores de los que se disponen son direcciones IP aisladas o detecciones y respuestas muy limitadas, los atacantes únicamente deberán cambiar dicho indicador, pero su manera de actuar será la misma. Sin embargo, si se tiene información más compleja y con capacidad suficiente para caracterizar el comportamiento de un atacante, no solo direcciones IP, hashes, dominios o redes, sino todo en su conjunto, se pueden establecer medidas que son lo suficientemente eficientes como para que el atacante tenga que cambiar su manera de proceder.

Las actividades de colaboración entre organizaciones y organismos internacionales se llevan a cabo mediante el intercambio de indicadores y toman forma en el concepto de compartición de inteligencia de amenazas. Para estar en el eslabón superior de esta pirámide estos indicadores deberán ser indicadores complejos. Las organizaciones han comprobado que intercambiar información sobre amenazas es necesario y fundamental dado el incremento de ataques desconocidos y su naturaleza dinámica. La importancia de compartir datos supone prevenir muchos ataques que están en su primera fase. En los sistemas existen comportamientos anómalos que muchas veces se descartan porque no suponen una amenaza por si solos. Compartiendo información se puede comprobar si una anomalía

que en un principio no supone un ataque, lo sea si se produce del mismo modo en varias organizaciones a la vez.

El análisis de la información transmitida en los indicadores será más eficiente si se basa en estándares. Para ello, se requiere el uso de indicadores estándar que sean comprendidos por todos. Los IoCs o Indicadores de Compromiso, se utilizan para describir de forma estandarizada las características técnicas de una amenaza y se caracterizan por ser estáticos y estar diseñados para luchar contra amenazas conocidas. Existen muchas iniciativas que especifican como se deben de documentar estos indicadores. IODEF y OpenIOC son algunos ejemplos.

Sin embargo, los indicadores IoC son limitados ya que son estáticos y se basan en caracterización de amenazas ya conocidas. Por ello, se han buscado a lo largo de los años otros que supongan soluciones más complejas, filtrando secuencias, patrones y comportamientos para poder anticipar posibles ataques.

STIX, CyBOX y TAXII son algunos de los estándares que están empezando a coger fuerza en el ámbito de la inteligencia de amenazas. STIX es un estándar para el intercambio de información estructurada sobre ciberseguridad e integra en su estructura recientemente la especificación de CyBOX. CyBOX especifica una serie de indicadores o ciberobservables para la caracterización de ciberamenazas. Estos estándares permiten caracterizar en sus estructuras, indicadores como IoCs, pero no solo ellos, sino también información sobre vulnerabilidades descritas en estándares como CVE.

El intercambio entre organizaciones es fundamental y, para ello, se necesita el uso de protocolos seguros que soporten dichos estándares. TAXII es un protocolo que permite intercambiar información CTI mediante HTTPS. Esta enfocado principalmente al intercambio de información STIX, pero puede transmitir a través de sus canales cualquier información relativa a inteligencia de amenazas.

Las plataformas de intercambio de ciberinteligencia como MISP también son una manera de recolectar información sobre amenazas y compartirla con distintas organizaciones. MISP es una plataforma abierta y completa que no solo recopila información sobre amenazas, sino que correla los datos que recibe y los almacena para después exportarlos. Si bien MISP soporta el formato de intercambio de información basado en STIX, MISP tiene un amplio soporte de formatos estructurados y una amplia cantidad de indicadores de seguridad. Basadas en MISP también se han implementado otras plataformas como REYES desarrollada por el CCN.

---

### 2.3.1 TECNOLOGÍAS PARA LA INTELIGENCIA DE AMENAZAS

---

A continuación, se detallan algunas de las herramientas mencionadas anteriormente que facilitan el intercambio de información de ciberinteligencia.

#### 2.3.1.1 CYBER OBSERVABLE EXPRESSION (CYBOX)

CyBOX es un lenguaje estandarizado para codificar y comunicar información de alta fidelidad sobre ciberobservables [11]. Un ciberobservable es un patrón que puede identificar un tipo de evento o estado relevante, observado en un dominio de ciberseguridad. CyBOX ofrece flexibilidad para dar una solución común para todos los casos de uso de ciberseguridad que requieran el tratamiento con ciberobservables. Por ello, CyBOX pretende ser lo suficientemente flexible como para definir patrones de ataques y perfiles *malware* con el fin de automatizar, compartir, detectar y analizar patrones lógicos que son evidencia de su existencia y permiten la gestión de riesgos en el sistema.

CyBOX ofrece a las diferentes organizaciones la capacidad de compartir información sobre posibles ataques mediante un formato estándar a través de indicadores. Cabe destacar que CyBOX se ha integrado a STIX. Los ciberobservables de CyBOX han pasado a llamarse Ciber Observables STIX y como se incorpora a STIX, es mantenido también por el grupo OASIS.

El formato CyBOX proporciona una amplia lista de objetos detallados y es independiente del fabricante a diferencia de otros formatos como OpenIoC que proporcionan poca flexibilidad.

### 2.3.1.2 STRUCTURED THREAT INTELLIGENCE EXPRESSION (STIX)

STIX es un lenguaje común y un formato usado para intercambiar información sobre inteligencia de ciberamenazas (CTI) [12]. STIX ofrece una gran expresividad en la creación de indicadores para la gestión e intercambio de información. No es una iniciativa separada del resto, sino que es una combinación de varias y ha sido creada de manera colaborativa con entidades significativas en el ámbito de la ciberseguridad, como son *MITRE Corporation* y el Comité Técnico sobre Inteligencia para Amenazas Informáticas OASIS.

STIX incluye en su definición la estructura de especificación de ciberobservables definidos anteriormente por CyBOX y que ahora se denominan, dentro del dominio de STIX, Ciber Observables STIX. Esto hace que a partir de esta nueva versión se hable de “STIX” como un único estándar para compartir la inteligencia de ciberamenazas. Además, STIX incluye información relacionada con vulnerabilidades, exploits, *malware* por lo que es compatible con modelos como CVE.

A fin de soportar una amplia variedad de casos de uso, STIX se caracteriza por una gran flexibilidad en el uso del estándar y por ser extensible, evitando el uso de sus características de carácter obligatorio siempre que sea posible.

La información en formato STIX se puede intercambiar utilizando distintos protocolos. OASIS ha definido el protocolo para el intercambio de datos estructurados en STIX TAXII (*Trusted Automated deXchange of Intelligence Information*) [13]. TAXII es un conjunto de protocolos y formatos para compartir información sobre ciberamenazas y está enfocado a compartir objetos STIX. Sin embargo, esto no implica que la información STIX no pueda ser compartida con otros tipos de protocolos.

STIX proporciona una arquitectura unificada que caracteriza la información sobre ciberamenazas. Esta arquitectura incluye:

- Ciberobservables (Ciber Observables STIX)
- Indicadores.
- Identidades.
- Técnicas, Tácticas y Procedimientos del atacante (incluyendo, *exploits*, *malware*, herramientas, infraestructuras, etc.)
  - Medidas.
  - Conjunto de Intrusión.
  - Análisis de *Malware*.
  - Campañas.
  - Actor de Amenaza.
  - Informe.
  - Opinión.
  - Notas.
  - Datos Observados.
  - Ubicación.

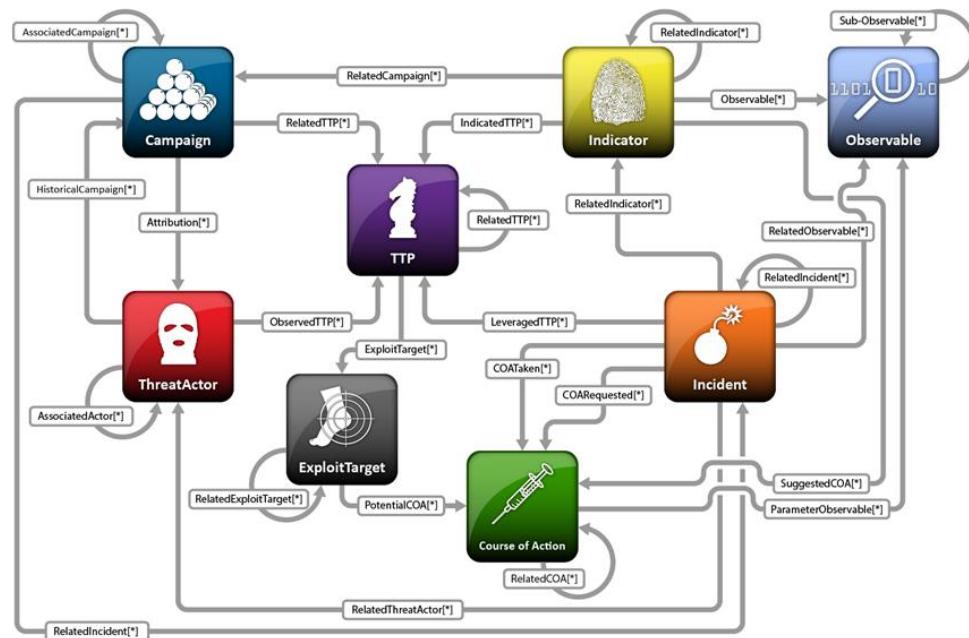


Figura 2. Representación de la arquitectura STIX [4]

STIX proporciona una representación global de objetos con grafos y es de código abierto y gratuito, lo que permite a las personas interesadas contribuir a su desarrollo. STIX se define en un lenguaje estandarizado basado en un formato XML, pero en su versión 2.1 fue actualizado al formato JSON.

Sin embargo, la adopción de STIX es demasiado reciente, lo que puede suponer desconfianza por parte de las organizaciones a la hora de utilizarlo.

STIX permite dar soporte de ciberamenazas, análisis y especificación de indicadores y patrones, responder a incidentes y compartir información sobre ciberamenazas. STIX parte de los anteriores casos de uso y proporciona un mecanismo para hacer frente a los ataques utilizando la inteligencia de ciberamenazas de una manera más eficiente, consistente, interoperable y con conciencia cibersituacional.

En la siguiente tabla se puede observar un resumen de las ventajas y desventajas que supone utilizar STIX frente a otros estándares menos recientes.

VENTAJAS	DESVENTAJAS
<b>IODEF</b> <ul style="list-style-type: none"> <li>○ Estándar definido por IETF e independiente de fabricante.</li> <li>○ Esquema XML.</li> <li>○ Mayor automatización del procesamiento de los datos.</li> <li>○ Facilidad de normalización para el intercambio de datos sobre incidentes con otras fuentes.</li> </ul>	<ul style="list-style-type: none"> <li>○ Adopción limitada.</li> <li>○ Puede contener información sensible que sea prohibida o complicada de compartir.</li> <li>○ Gran granularidad.</li> </ul>

OpenIOC	
<ul style="list-style-type: none"> <li>○ Es gratuito, bajo la licencia Apache 2.</li> <li>○ Esquema XML.</li> <li>○ Dispone de un software libre para crear indicadores OpenIOC.</li> </ul>	<ul style="list-style-type: none"> <li>○ Está limitado a los productos del fabricante.</li> <li>○ Poca flexibilidad.</li> <li>○ No soporta TTPs.</li> </ul>
STIX	
<ul style="list-style-type: none"> <li>○ Integra CyBOX.</li> <li>○ Es de código abierto y gratuito.</li> <li>○ Formato XML, actualizado al formato JSON.</li> <li>○ Integración de otros esquemas.</li> <li>○ Creado de forma colaborativa.</li> </ul>	<ul style="list-style-type: none"> <li>○ Adopción reciente</li> </ul>

Tabla 1. Comparación de STIX con otros estándares

### 2.3.1.3 HERRAMIENTAS DE COMPARTICIÓN DE AMENAZAS: MISP

MISP es una plataforma cuyo principal objetivo es intercambiar información entre diferentes organizaciones que generan ciberinteligencia [14]. De esta manera, las organizaciones participantes en esta plataforma serán capaces de compartir y recibir información sobre ciberinteligencia en distintos formatos. Varios CSIRTs europeos y privados han creado una plataforma de intercambio utilizando MISP, lo que ha facilitado su expansión.

MISP es compatible con estructuras STIX y otros formatos, como OpenIOC. Es un software gratuito y de código abierto. No solo es una plataforma para compartir información, sino que almacena y establece relaciones de correlación entre distintos IoCs y otros indicadores, y recoge información tanto técnica como no técnica, como información sobre *malware* y ataques ya detectados. A pesar de su alta adopción, MISP no implementa TAXII como protocolo de transporte de intercambio de la información de ciberinteligencia.

MISP pretende establecer una arquitectura federada de forma segura y controlada. Son muchos los proyectos que se han basado en MISP para su implementación, entre ellos destaca REYES, desarrollado por el CCN para compartir información exclusiva de ciberamenazas a través de un portal centralizado.

### **3 ONTOLOGÍAS: MODELOS FORMALES DE REPRESENTACIÓN DE LA INFORMACIÓN**

La gran variedad de sistemas de seguridad ayuda a defender una organización frente a las amenazas. Sin embargo, esta multitud de herramientas puede turbar su objetivo y al comprender una gran cantidad de información sobre seguridad en distintos aspectos, la manera de administrarla y entenderla se vuelve complicada y compleja. Es por ello por lo que surgen unos sistemas capaces de incorporar toda esa información siguiendo una estructura para que su gestión no se vuelva una tarea difícil. Estos sistemas son los sistemas de ontologías, la base de este trabajo.

El concepto de ontologías surge en el marco de la ingeniería del conocimiento y de la Web. Una ontología es una representación formal del conocimiento en la que se definen conceptos, propiedades y relaciones entre los mismos. Una ontología ofrece una manera de limitar la complejidad de la información y organizarla para que sea entendida en su dominio.

La Web Semántica pretende representar el conocimiento de forma que sea legible e interpretado por máquinas. De ahí, la especificación de Web “Semántica”. Con esta idea, Tim Berners-Lee pretendía organizar mejor la información que se subía a las páginas web y así ofrecer búsquedas más precisas por significado y no por contenido. Las máquinas serían capaces de interpretar y gestionar el conocimiento que se guardaba en las páginas web.

Las ontologías están formadas por dos elementos, una taxonomía y un conjunto de reglas de inferencia. La taxonomía es la definición de clases y subclases de objetos y las relaciones existentes entre dichos objetos. También se encarga de la asignación de propiedades a las clases. Las reglas de inferencia añaden una capacidad adicional útil para el usuario humano que hace uso de la ontología. Estas reglas permiten expresar restricciones y condiciones sobre los objetos de las clases definidas mediante la taxonomía. La forma de expresar restricciones y reglas en un lenguaje de ontologías es mediante axiomas. Todas estas reglas pueden utilizarse para realizar un razonamiento que infiera nuevo conocimiento y enriquezca la ontología.

Si bien el uso de ontologías ha ganado fuerza con la aparición de la Web Semántica, actualmente las ontologías se utilizan en muchos otros ámbitos como la medicina, industria o ciberseguridad. Es este último el que aquí nos atañe. La creación de una ontología permitirá estructurar la información de una organización para representar formalmente cada uno de sus componentes (activos, vulnerabilidades, amenazas, etc.), con el fin de automatizar el procesamiento de esa información y generar nuevo conocimiento. Una vez procesada la información, se podrán deducir amenazas y riesgos y generar a partir de ellos otros nuevos, así como calcular el riesgo de la organización y/o establecer estrategias para el tratamiento de los mismos.

Existen varios lenguajes para la definición de ontologías que permiten crear el modelo formal con el que se pretende representar el entorno de conocimiento. Además, gracias a las ontologías, se puede agrupar el conocimiento, procedente de distintas partes, en un marco común para producir inteligencia, en este caso inteligencia de amenazas. Los lenguajes de ontologías se caracterizan principalmente por los siguientes requisitos:

- Una sintaxis bien definida.
- Una semántica formal.
- Ser un soporte como para poder realizar razonamientos eficientes.
- Expresividad.
- Conveniencia de expresión.

La sintaxis de los lenguajes de ontologías debe de estar bien definida puesto que se debe poder expresar adecuadamente cualquier elemento de información para que sea procesado por máquinas correctamente. La semántica formal explicita que no cabe lugar a interpretaciones subjetivas y está basada en la lógica matemática. A partir de esta semántica formal se podrán realizar los razonamientos adecuados, tales como clasificación, consistencia de la ontología o equivalencia de clases. Además, si

estos razonamientos se realizan automáticamente y por máquinas se podrán comprobar más casos de usos que si se hiciese manualmente y, por tanto, será posible diseñar, integrar y compartir ontologías más largas y complejas.

Sin embargo, cuanto más expresivo es un lenguaje, menos eficiente es el razonamiento que se puede hacer a partir de él. Por eso, debe de haber un equilibrio entre la expresividad de un lenguaje y la eficiencia de un razonamiento. Los lenguajes de ontologías deben apoyarse en otras herramientas para poder interpretar la ontología e inferir nuevo conocimiento a partir de los conceptos y relaciones que en ella se definen. Las reglas de comportamiento y los razonadores semánticos aportan esa capacidad para enriquecer la base de conocimiento.

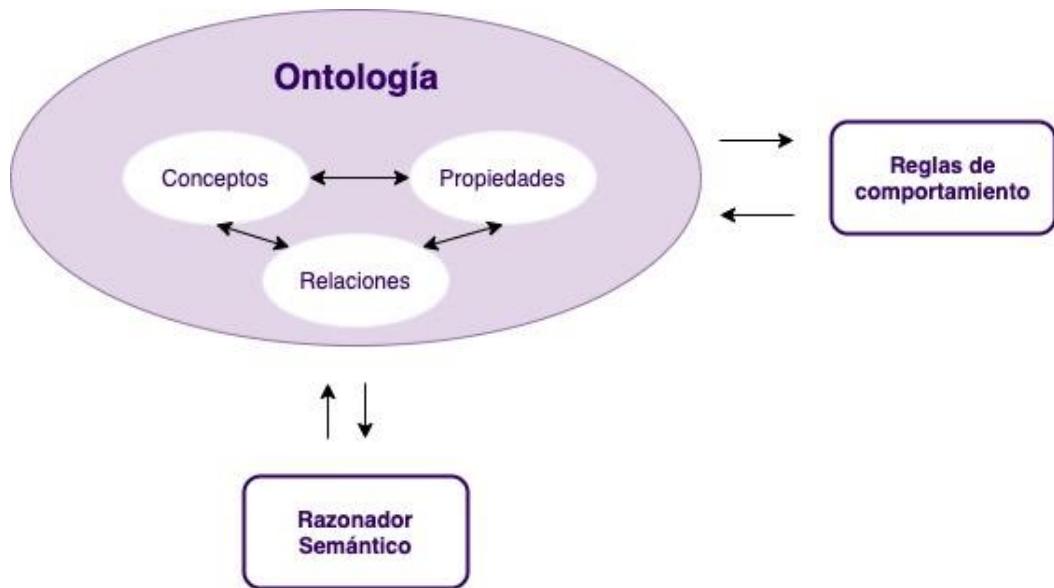


Figura 3. Sistema de ontologías

Existen numerosos lenguajes orientados a la definición de ontologías, que se clasifican en diferentes paradigmas según sus patrones de especificación de información, la terminología utilizada en cada uno de ellos, los elementos que incorporan y el uso que se hace de ellos. Antes de que surgiese la Web Semántica, ya existían lenguajes, como SHOE, que permitían realizar un análisis inteligente en la web mediante la definición de ontologías sobre documentos HTML, de manera que fueran legibles por máquinas.

Con la Web Semántica surgen los lenguajes estándar RDF y RDFS que son desarrollados por el *World Wide Web Consortium* (W3C) para representar ontologías. La mayor parte de la Web Semántica está basada en estos lenguajes. RDF y RDFS son lenguajes basados en las llamadas tripletas de datos estructuradas según la forma: *sujeto + predicado + objeto*. Sin embargo, debido a su simplicidad, estos lenguajes proporcionan poca expresividad y una pobre inferencia de conocimiento.

Por ello, aparecen otros lenguajes que surgen de la extensión de los anteriores, como DAML, que más tarde se extiende en DAML+OIL, está basado en estándares de W3C y proporciona mayor inferencia que RDFS, pero se convierte en un lenguaje bastante complejo y poco flexible. De la extensión de estos lenguajes surge el desarrollo de otros, OWL y OWL 2, más avanzados y que aportan un vocabulario más complejo para describir los conceptos y sus propiedades. A pesar de que la estructura de estos lenguajes es muy similar, OWL2 tiene algunas ventajas ya que ofrece mayor expresividad y es soportado por un mayor número de razonadores semánticos.

Las reglas de comportamiento son lenguajes que describen restricciones, reglas y políticas que permiten expresar de manera formal comportamientos de cada una de las entidades que constituyen las ontologías. Las restricciones se representan con axiomas. Un axioma es una sentencia que especifica

las relaciones que deben cumplir los elementos de la ontología. OWL2 integra muchos tipos de axiomas. Sin embargo, se necesitan las reglas de comportamiento para crear reglas que permitan generar nuevo conocimiento y expresar más restricciones que los lenguajes de ontologías.

Existen distintos tipos de lenguajes de definición de reglas que posteriormente se detallan. Entre ellos, destaca SWRL. SWRL es el lenguaje de definición de reglas más utilizado en la Web Semántica y permite escribir reglas expresadas en términos de conceptos OWL. Ofrece una gran expresividad y define restricciones que no son posibles con OWL o RDF. Antes de este lenguaje, existían otros lenguajes de reglas en los que se ha basado SWRL, como es el lenguaje RuleML usado también en la Web Semántica. Además, el ámbito de los lenguajes de reglas ha ido evolucionado y se han creado lenguajes basados en Java, como Jess, que facilitan su integración.

Sin embargo, las reglas y los lenguajes de ontologías no son suficientes para crear un modelo formal. Por ello, se necesita un razonamiento semántico. El razonamiento semántico se consigue, valga la redundancia, mediante el uso de razonadores semánticos, que no son más que un software capaz de inferir conocimiento a partir de un conjunto de axiomas y reglas de comportamiento. Los razonadores verifican la correcta consistencia de la ontología, tanto de los conceptos y propiedades, como de las reglas que en ella se definen, llevando a cabo un razonamiento sobre la semántica de la ontología. Es por ello por lo que los razonadores hacen uso de algoritmos y se basan en la lógica matemática.

Existen muchos razonadores semánticos. Las diferencias entre ellos radican principalmente en los algoritmos de inferencia que usan para razonar y obtener nuevo conocimiento, en los lenguajes de definición de ontologías y de definición de reglas que soportan, y en otras características como su lenguaje de implementación o expresividad.

Los razonadores se dividen en distintas generaciones en función del año de publicación. Un razonador será más eficiente cuando mantenga un equilibrio entre una alta cantidad de funcionalidades soportadas (como fuerte inferencia, soporte web, consultas mediante *queries*, multiusuario...) y un tiempo de respuesta e inferencia bajo. La elección de un razonador u otro depende también de las necesidades del proyecto. Pellet es uno de los razonadores más eficientes y rápidos y es compatible con la mayor parte de los lenguajes de definición de ontologías y reglas.

La combinación de los lenguajes de ontologías, las reglas y el razonamiento semántico permite crear sistemas estructurados que son legibles por máquinas y que facilitan la gestión de grandes cantidades de información, así como la automatización de funciones que manualmente resultan complejas.

A continuación, se realiza un análisis de algunos de los diferentes lenguajes de definición de ontologías, lenguajes de definición de reglas y razonadores semánticos existentes que son de gran relevancia para el trabajo. Al final de cada apartado se representan en una tabla las ventajas y desventajas que aporta cada uno de ellos y que posteriormente facilita la justificación de los utilizados para este proyecto.

### 3.1 LENGUAJES DE DEFINICIÓN DE ONTOLOGÍAS

#### 3.1.1 RESOURCE DESCRIPTION FRAMEWORK (RDF)/RDF-SCHEMA (RDFS)

En la Web Semántica, los lenguajes RDF y RDFS son los lenguajes estándar desarrollados por W3C para la representación de metadatos y ontologías [15] [16]. Debido a que RDF fue el primer lenguaje con semántica formal comprendido por máquinas, la mayor parte de la Web Semántica está basada en el mismo.

El lenguaje RDF está basado en triplets de datos formadas por: *sujeto + predicado + objeto*. El *sujeto* es el concepto que se explica, el *predicado* es la relación o propiedad que se quiere establecer entre el sujeto y el *objeto*, y este último, es el valor de la propiedad. Cada uno de ellos es un recurso y se identifica con un URI, *Uniform Resource Identifier*. El conjunto de las triplets dentro de un dominio forma un grafo RDF que se puede entender como una ontología RDF.

La sintaxis formal permite razonar e inferir nuevo conocimiento. Sin embargo, RDF muestra una pobre inferencia y supone una limitación debido a su simple sintaxis y a la poca expresividad que aportan las tripletas.

Para superar estas limitaciones se desarrolla el lenguaje RDFS que extiende RDF y aporta una mayor capacidad de inferencia de conocimiento. RDFS permite ser utilizado junto con RDF y, además, define restricciones sobre los recursos y las propiedades RDF. Análogamente a RDF, RDFS permite visualizar los conceptos y las relaciones mediante un grafo, pero en este caso, un grafo RDFS.

---

### 3.1.2 ONTOLOGY INTERCHANGE LANGUAGE (OIL)

---

OIL es un lenguaje de definición de ontologías que pertenece a la Web Semántica y ha sido desarrollado por W3C. Está basado en el lenguaje XML y fue definido como una extensión de RDFS, por lo que es compatible con él.

OIL combina los modelos de lenguajes basados en etiquetas con una semántica formal y la capacidad de razonamiento proporcionados por la lógica descriptiva mediante axiomas. OIL está formado por varias capas cada una perteneciente a un sub-lenguaje, entre las que cabe destacar RDFS. Cada capa va añadiendo una nueva funcionalidad.

Sin embargo, OIL se ve limitado por la falta de expresividad a la hora de la declaración de los axiomas.

---

### 3.1.3 DAML + OIL

---

Este lenguaje surge de la unión del lenguaje europeo llamado OIL y de la propuesta americana DAML, lenguaje que fue desarrollado por el grupo de trabajo DARPA (*Defense Advanced Research Projects Agency*). DAML+OIL está basado en estándares de W3C, y mantiene algunas ideas del lenguaje OIL, pero potenciando la lógica descriptiva.

El lenguaje XML es un lenguaje limitado en lo que respecta a describir relaciones entre objetos usando etiquetas y más aún si se describen ontologías completas. El lenguaje DAML surge de la extensión del lenguaje XML y RDF. DAML permite definir de manera más precisa ontologías y ofrecer un lenguaje con mayor expresividad y entendible por máquinas. DAML es más potente que RDFS y supone un avance ya que ofrece una mayor capacidad de inferencia que sus predecesores.

A partir de su desarrollo surgen otros lenguajes más avanzados como OWL, que añaden un vocabulario más expresivo para definir las clases y sus propiedades.

---

### 3.1.4 ONTOLOGY WEB LANGUAGE (OWL)

---

OWL es un lenguaje diseñado para representar conocimiento complejo sobre conceptos y las relaciones entre los mismos. Es el lenguaje más utilizado en la Web Semántica para la representación de ontologías explícitamente [17].

OWL reemplaza a DAML+OIL como el estándar de W3C para la representación de ontologías. A pesar de sus similitudes sintácticas y semánticas, OWL es mucho más complejo ya

que añade más vocabulario para describir propiedades y conceptos. También es una extensión de RDF y tiene mayor expresividad que este.

OWL es un lenguaje de ontologías orientado a objetos. Existen distintas sintaxis de las que hace uso OWL para poder intercambiar las ontologías entre diferentes herramientas, tales como RDF/XML, OWL/XML o Manchester OWL. La primera está basada en la representación XML de grafos RDF, permitiendo mapear directamente entre la sintaxis OWL y la de RDF. La segunda está basada en una sintaxis funcional y es más sencilla que la anterior. La sintaxis Manchester para OWL es muy compacta y sencilla de leer y escribir y fue diseñada principalmente para la integración sencilla del lenguaje OWL en editores como Protégé.

La semántica formal en OWL es más precisa ya que OWL permite definir clases y subclases y completa a RDF/RDFS definiendo propiedades semánticamente más avanzadas, definiendo axiomas que se aplican sobre los elementos para restringir su significado y relaciones más detalladas mediante propiedades transitivas, clases disjuntas, etc. La semántica formal se garantiza a través del mapeado de OWL con formalismos propios de la lógica de predicados y la lógica descriptiva, un campo de investigación que ha estudiado las lógicas que constituyen el fundamento formal de OWL.

OWL está estructurado en distintas capas que pueden ser adaptadas según las necesidades de cada usuario. La decisión de utilizar una u otra depende del nivel de expresividad requerido y de los diferentes tipos de aplicaciones que existen. Por tanto, se han desarrollado tres sub-lenguajes para cubrir esas características:

- **OWL Lite:** es la versión más simple y se utiliza para expresar de manera sencilla la ontología y restricciones simples. Se define como un subconjunto del resto y ha sido diseñada para principiantes o aquellos que abogan por la simplicidad. No es compatible con RDF/RDFS, pero su semántica proviene de un subconjunto de RDFS.
- **OWL DL:** soporta una gran expresividad, mientras mantiene la integridad computacional (posibilidad de inferir nuevas conclusiones a partir de la información existente) y la capacidad de decisión. OWL DL incluye todas las construcciones de lenguaje OWL, pero se deben usar bajo ciertas restricciones. OWL DL no es compatible con ontologías que usan la máxima expresividad de RDF/RDFS. El nombre de OWL DL se debe a su correspondencia con la lógica descriptiva.
- **OWL Full:** es la versión más amplia y ha sido creada para aquellos usuarios que prefieren la máxima expresividad. Es una mezcla entre OWL Lite y OWL DL y fue creada para ser compatible con RDF *Schema*. Los modelos basados en OWL Full pueden usar construcciones RDF, RDFS y OWL. Sin embargo, no se garantizan las propiedades computacionales, ya que es poco probable que cualquier software de razonamiento sea capaz de obtener un razonamiento completo para cada característica de OWL Full.

OWL Full puede ser considerada como una extensión de RDF, mientras que OWL Lite y OWL DL son una extensión restringida de RDF.

Como ya se ha mencionado, el lenguaje RDF está formado por un conjunto de tres entidades que representan los datos semánticos forma de expresiones sujeto-predicado-objeto. Cada una de estas partes se puede direccionar con un URI único de la forma: <http://www.semanticweb.org/ontologies/2020/0#Casa>. Gracias a esta representación se pueden hacer consultas y razonamientos sobre ellos sin ambigüedad. Sin embargo, alguien podrá definir tripletes que no tengan sentido, y RDF sabrá que semánticamente no es correcto, pero no puede hacer nada para validar lo que se ha escrito. Esto puede provocar que la definición de ontologías no válidas.

Con el lenguaje OWL se define lo que se puede escribir con RDF para crear una ontología correcta. OWL consigue mantener un equilibrio entre la alta expresividad y la capacidad de soportar

razonamientos eficientes. A pesar de que han surgido otros lenguajes, OWL sigue siendo uno de los lenguajes más utilizados para la representación de ontologías.

### 3.1.5 OWL 2

OWL2 [18] es una extensión y revisión de OWL desarrollada por W3C. Al igual que OWL, su objetivo es facilitar el desarrollo de ontologías y el intercambio de información en la Web haciendo que el contenido de la Web Semántica sea más accesible e interpretable por máquinas.

OWL 2 extiende la sintaxis y la semántica de OWL y su expresividad es mayor que en OWL. OWL2 suele utilizar como sintaxis para el intercambio de información RDF/XML al igual que OWL, pero también soporta otras como la sintaxis Turtle de RDF, la sintaxis Manchester, OWL/XML, etc. En cuanto a la semántica, existen dos modelos: semántica directa, conocida como OWL-DL o sintaxis funcional y la semántica basada en RDF, conocida como OWL Full. Ambas alternativas son usadas por los razonadores y otras herramientas para comprobar la consistencia de las clases y las instancias.

Al igual que su predecesor, OWL2 define diferentes perfiles o sub-lenguajes, cada uno de los cuales proporciona una determinada eficiencia del razonamiento según distintos escenarios. Se diferencian por las propiedades computacionales y lógicas que garantizan. Derivan del lenguaje OWL-DL:

- **OWL-EL:** permite trabajar con ontologías de gran tamaño donde es posible intercambiar potencia expresiva por eficiencia en el razonamiento.
- **OWL-QL:** permite realizar consultas a la ontología por medio de lenguajes como SQL, por lo que es útil cuando se desea crear ontologías que luego se pretende consultar como si fuesen bases de datos.
- **OWL-RL:** implementa un razonamiento haciendo uso de las tecnologías de bases de datos directamente sobre triplets RDF, por lo que se usa principalmente en aquellas aplicaciones en la que es útil trabajar directamente sobre triplets RDF.

Elegir uno u otro depende de la ontología a representar y del objetivo del razonamiento.

Como OWL2 extiende OWL, OWL2 tiene más funcionalidades que mejoran la expresividad y soporta un mayor número de razonadores semánticos que los anteriores lenguajes.

En la siguiente tabla se puede observar un resumen de las ventajas y desventajas de utilizar cada uno de los lenguajes anteriores:

VENTAJAS	DESVENTAJAS
<b>RDF/RDFS</b>	<ul style="list-style-type: none"> <li>○ Lenguajes más difundidos en la Web Semántica.</li> <li>○ Gracias a OWL muchas deficiencias se han mejorado.</li> <li>○ RDFS añade más expresividad.</li> </ul> <ul style="list-style-type: none"> <li>○ Expresión en triplets</li> <li>○ Visualización no intuitiva</li> <li>○ Lenguaje muy primitivo</li> <li>○ No permite restricciones.</li> <li>○ No permite cardinalidad.</li> <li>○ Razonamiento pobre.</li> </ul>
<b>DAML+OIL</b>	

<ul style="list-style-type: none"> <li>○ Extiende RDF/RDFS.</li> <li>○ Mayor potencia de inferencia que RDFS.</li> <li>○ Mayor expresividad que RDF y RDFS</li> </ul>	<ul style="list-style-type: none"> <li>○ Lenguaje bastante complejo</li> </ul>
<b>OWL</b>	
<ul style="list-style-type: none"> <li>○ Mayor vocabulario para la descripción de clases y propiedades.</li> <li>○ Más potente y expresivo que RDF, RDFS y DAML+OIL.</li> <li>○ Completa a RDF/RDFS definiendo relaciones entre clases semánticamente más avanzadas y complejas.</li> <li>○ Ofrece mayor eficiencia de razonamiento.</li> </ul>	<ul style="list-style-type: none"> <li>○ Falta de constructores, como en las restricciones de cardinalidad o tipos de datos que limitan la expresividad y que da lugar diseños poco óptimos.</li> <li>○ No se puede validar el sublenguaje en el que está escrita una ontología.</li> </ul>
<b>OWL2</b>	
<ul style="list-style-type: none"> <li>○ Extiende la semántica y la sintaxis de OWL.</li> <li>○ Mayor expresividad que OWL.</li> <li>○ Soporta mayor número de razonadores semánticos.</li> </ul>	<ul style="list-style-type: none"> <li>○ Adopción reciente</li> </ul>

Tabla 2. Comparación de los lenguajes de definición de ontologías

## 3.2 LENGUAJES DE DEFINICIÓN DE REGLAS DE COMPORTAMIENTO

### 3.2.1 JESS

*Java Expert System Shell* (Jess) es un potente motor de reglas y lenguaje de scripts desarrollado en Java [19]. Jess se inspiró en el lenguaje CLIPS basado en C para su desarrollo. Es un motor de reglas ligero y uno de los más rápidos. Jess se puede usar tanto como lenguaje de definición de reglas, como para la inferencia de conocimiento a partir de ellas.

Dado que está desarrollado en Java, permite el acceso a todas las APIs de Java, crear y manipular objetos de Java y su filosofía de funcionamiento es actuar en respuesta a entradas.

Jess es un lenguaje multiplataforma, permite su desarrollo a través de la plataforma de Eclipse, puede trabajar conjuntamente con editores como Protégé, a través del plugin JessTab, es utilizado ampliamente en sistemas expertos y es fácil de integrar con cualquier aplicación. Sin embargo, Jess es poco eficiente y no existe mucha documentación sobre las posibilidades de su uso.

### 3.2.2 RULEML

*Rule Markup Language* (RuleML) es un lenguaje estándar de especificación de reglas para la definición, publicación e intercambio de reglas en la Web Semántica [20]. RuleML unifica los lenguajes de reglas serializados en XML. El lenguaje RuleML está basado en el sub-lenguaje Datalog (intersección entre SQL y PROLOG) y usa sintaxis XML para la definición de las reglas.

Las reglas que pueden ser definidas utilizando RuleML son de diferentes tipos, entre los que se puede destacar las siguientes: reglas de producción (implicaciones if-then), reglas reactivas (evento-condición-acción), reglas de integración y derivación, etc.

Sin embargo, RuleML tiene una gran desventaja: no soporta el uso de forma conjunta con OWL y OWL2.

### 3.2.3 R2ML

*REWERSE Rule Markup Language* (R2ML) ha sido desarrollado por el grupo de trabajo REWERSE para el intercambio de reglas entre sistemas y herramientas [21]. R2ML es un lenguaje de reglas basado en XML y soporta reglas de integridad, reglas de derivación, reglas de producción y reglas de reacción.

R2ML ayuda al intercambio e integración de reglas entre distintos lenguajes como los lenguajes Datalog (como SWRL) y entre herramientas software específicas. Permite publicar y desarrollar reglas en una red, además de inferir resultados de cualquier sistema.

### 3.2.4 SWRL

*Semantic Web Rule Language* (SWRL) es un lenguaje de especificación de reglas en la Web Semántica [22]. SWRL está basado en OWL y RuleML y es una combinación de los sub-lenguajes de OWL, OWL DL y OWL Lite con los sub-lenguajes *Unary/binary* Datalog de RuleML.

Mediante el lenguaje SWRL se describen restricciones o axiomas de alto nivel, que permiten crear nuevo conocimiento y expresar restricciones más complejas que los lenguajes de ontologías. Los axiomas que aquí se definen, permiten especificar las llamadas reglas.

Una regla es un axioma formado por un antecedente y un consecuente, cada uno de los cuales está compuesto por una agrupación de átomos.



De esta manera, si se cumplen las condiciones especificadas en los átomos del antecedente se ejecutarán las condiciones especificadas en el consecuente. Esto supone una relación de implicación entre ambos, antecedente y consecuente y, de este modo, los datos que aparecen en el antecedente pueden aparecer en el consecuente. Esto conlleva a una relación de seguridad, ya que solo se podrán sacar conclusiones sobre la información presente.

Los átomos son predicados de la forma  $C(x)$ ,  $P(x, y)$ ,  $sameAs(x, y)$ ,  $differentFrom(x, y)$  o  $builtin(r, x, ...)$ ; donde  $C$  es una clase o concepto definida en OWL,  $P$  es una propiedad definida en OWL,  $r$  es una relación *built-in* y  $x$  e  $y$  representan variables o ejemplares en OWL/OWL2 como valores de las clases o conceptos. A través de estos átomos se pueden especificar, entre otras cosas: pertenencia a una clase, una relación entre dos variables o una relación entre un concepto y un literal. Los *built-ins* de SWRL permiten realizar operaciones sobre los distintos tipos de datos como comparaciones, operaciones matemáticas, operaciones con listas o negaciones lógicas, entre otros.

Por tanto, añadiendo las reglas SWRL sobre una ontología se puede incluir información de forma implícita. Esta información puede que realmente no sea nueva y sirva para comprobar que ciertos axiomas o restricciones se cumplen. O, por el contrario, puede ser información nueva y aporta conocimiento adicional sobre la ontología. Esta información no se representa de forma explícita en la ontología, y para verla de esta manera sería necesario generar otra ontología inferida a partir de la anterior.

SWRL extiende el conjunto de axiomas de OWL para ofrecer mayor expresividad a la especificación de las reglas, lo que permite incluir axiomas de reglas llamados cláusulas de Horn (reglas condicionales). Utiliza el lenguaje Datalog para definir las cláusulas Horn en una base de conocimiento OWL. SWRL es un lenguaje que permite escribir reglas que pueden ser expresadas en términos de conceptos OWL para ofrecer capacidades de razonamiento deductivo mucho más potentes que el propio lenguaje OWL.

SWRL se caracteriza ser compatible con OWL, RDF y XML, es fácil de integrar con otras herramientas como Protégé a través de *plugins* y es el lenguaje recomendado para la definición de reglas por la Web Semántica.

Cabe destacar, que, a pesar de sus múltiples ventajas, SWRL es limitado al estar destinado para clases OWL y predicados binarios; y no permite especificar metainformación, la cual es útil para especificar prioridades, reglas remotamente y analizar políticas y resolución de conflictos.

En la siguiente tabla se puede observar un resumen de las ventajas y desventajas de utilizar cada uno de los lenguajes anteriores:

	VENTAJAS	DESVENTAJAS
<b>Jess</b>		
	<ul style="list-style-type: none"> <li>○ Escrito en Java</li> <li>○ Su desarrollo en Java uso permite el uso APIs de Java e integrarlo fácilmente con otras aplicaciones.</li> </ul>	<ul style="list-style-type: none"> <li>○ Licencia especial para aplicaciones comerciales</li> <li>○ Poco eficiente</li> <li>○ Poca documentación</li> </ul>
<b>RuleML</b>		
	<ul style="list-style-type: none"> <li>○ Ofrece muchos tipos de reglas como reglas tipo if-then, de derivación, de integridad, etc.</li> </ul>	<ul style="list-style-type: none"> <li>○ No permite el uso con lenguajes OWL y OWL2</li> </ul>
<b>R2ML</b>		
	<ul style="list-style-type: none"> <li>○ Define reglas de reacción.</li> <li>○ Usado para la integración de reglas con otros lenguajes como SWRL.</li> <li>○ Integración e intercambio de reglas entre herramientas software específicas</li> </ul>	<ul style="list-style-type: none"> <li>○ Compleja para ser soportada por una herramienta que trabaje eficientemente</li> </ul>
<b>SWRL</b>		
	<ul style="list-style-type: none"> <li>○ Compatible con OWL y OWL2, RDF y XML.</li> <li>○ Incluye axiomas de reglas llamados cláusulas Horn.</li> <li>○ La funcionalidad de SWRL puede extenderse mediante <i>built-ins</i>.</li> </ul>	<ul style="list-style-type: none"> <li>○ limitado al estar destinado para clases OWL y predicados binarios</li> <li>○ No especifica metainformación.</li> </ul>

Tabla 3. Comparación de los lenguajes de definición de reglas

### 3.3 TECNOLOGÍAS PARA EL MANEJO Y GESTIÓN DE ONTOLOGÍAS Y REGLAS DE COMPORTAMIENTO

En los apartados anteriores se ha realizado un estudio de los distintos lenguajes existentes para definir ontologías y de los razonadores y lenguajes de definición de reglas que permiten completar la información que se representa en una base de conocimiento.

Cada uno de estos elementos son esenciales para crear una representación formal sobre un dominio determinado y permiten estructurar la información de tal manera que sea procesada por máquinas automáticamente.

Ahora bien, utilizar estos lenguajes requiere de un conocimiento avanzado a bajo nivel en programación. Por ello, existen muchas herramientas que facilitan la programación de las ontologías y hacen más sencillo el uso de razonadores semánticos y reglas.

Si bien existe un gran número de herramientas para el manejo y gestión de ontologías, en este apartado se estudian solo algunas que resultan relevantes y de gran ayuda para realizar el trabajo de fin de máster.

---

#### 3.3.1 EDITOR PROTÉGÉ

---

El editor Protégé es una plataforma de código abierto desarrollada por una comunidad sólida de desarrolladores y equipos corporativos[23]. Protégé permite construir ontologías, generar y ejecutar reglas y utilizar razonadores mediante plugins. Las soluciones de Protégé son muy utilizadas en muchos ámbitos como la biomedicina o modelos corporativos.

Protégé es un editor de ontologías implementado en Java y soporta XML Schema, RDF y OWL y cuenta con un ambiente “plugin-and-play”. Tiene soporte para editar, importar y exportar múltiples ontologías y ofrece interfaces para la conexión con razonadores como Pellet, FaCT++ o HermiT.

Dado que la programación de ontologías de grandes dimensiones puede resultar complicado, Protégé ofrece un interfaz fácil de utilizar, configurable por el usuario que puede añadir tablas o vistas en función de sus necesidades. Ofrece una serie de herramientas configurables mediante plugins para mejorar su funcionalidad, soporte de reglas SWRL mediante SWRLTab para la definición de reglas, permite realizar consultas mediante SPARQL y ofrece plugins que permiten visualizar la ontología de manera gráfica, como OWLViz o OntoGraf.

---

#### 3.3.2 OWL API

---

OWL API es un *framework* implementado en Java para crear, manipular y serializar ontologías OWL [24]. Es de código abierto y mantenido por un conjunto de grupos corporativos como Clark & Parsia LLC y la Universidad de Manchester.

La implementación de OWL API está basada en la idea de proporcionar una visualización a “alto nivel” para que los desarrolladores no tengan que especializarse o tener conocimiento sobre la serialización de estructuras de datos, dada la intima relación entre OWL y las estructuras RDF. Por ello, esta API está basada en el lenguaje OWL.

OWL API soporta la inferencia de conocimiento gracias a la existencia de razonadores que escuchan los cambios de la ontología y los procesan para verificar su consistencia, clasificar las clases en una jerarquía, etc. Entre las implementaciones de razonadores en la OWL API se pueden encontrar razonadores como FaCT++, HermiT o Pellet que se integran fácilmente. Además, OWL API también soporta la implementación y ejecución de reglas SWRL complejas.

Además, como está basado en OWL soporta diferentes tipos de sintaxis como RDF/XML, OWL/XML, Manchester, Turtle, etc., que ofrecen diferentes características que optimizan el proceso de creación de ontologías según las necesidades.

Las ultimas versiones de OWL API se centran en OWL 2 y es uno de los framework que ofrece más utilidades y más fácil de usar e implementar dado su nivel de abstracción.

---

### 3.3.3 JENA

---

Jena es un framework implementado en Java para la construcción de ontologías [25] . Es un entorno de código abierto creado para manejar datos RDF, no solo ontologías basadas en RDF. Además, soporta no solo RDF, sino también RDFS y OWL y permite ejecutar inferencias a través de reglas y razonadores.

Jena es una de las APIs más utilizadas para entornos que utilizan el lenguaje OWL y RDF, y ofrece servicios para modelar ontologías, representar modelos, crear bases de datos mediante grafos RDF y consultas mediante el procesador SPARQL. El soporte para la inferencia de conocimiento en Jena ofrece la posibilidad de utilizar los que Jena tiene por defecto, pero también permite añadir otros nuevos.

Cabe destacar que Jena está basada en RDF por lo que gracias a ello se puede decir que es una de las APIs para el manejo de datos RDF y OWL más extensa y que permite, a partir de RDF, crear axiomas y restricciones en OWL, puesto que OWL está basado en el lenguaje RDF.

---

### 3.3.4 RAZONADORES SEMÁNTICOS

---

Las reglas y el lenguaje de definición de ontologías no son suficientes para crear un modelo formal. Por ello, se necesitan otros componentes para verificar la correcta consistencia de la ontología, tanto de los conceptos y propiedades, como de las reglas que en ella se definen.

Un razonador semántico es un software capaz de inferir conocimiento a partir de un conjunto de axiomas y reglas de comportamiento y mantener su consistencia. Las reglas de comportamiento se determinan mediante los lenguajes especificados en el apartado anterior.

Muchos razonadores hacen uso de la lógica de predicados para razonar e inferir conocimiento. La lógica de predicados evalúa los conceptos de las ontologías y las relaciones entre ellos. Esta lógica permite verificar afirmaciones y realizar deducciones de los hechos, con el fin de inferir nuevo conocimiento. Esta inferencia suele ser realizada mediante encadenamiento hacia delante o hacia atrás (*forward o backward chaining*). El encadenamiento hacia delante comienza por los hechos conocidos para derivar conocimiento válido, mientras que el encadenamiento hacia atrás comienza con un hecho en particular y lo verifica encontrando todas las soluciones posibles.

Existen muchos razonadores capaces de inferir conocimiento. Las diferencias entre ellos se deben principalmente a las técnicas que usan para razonar y obtener nuevo conocimiento, a los lenguajes de definición de ontologías y de definición de reglas que soportan, además de otras características adicionales que determinan su eficiencia.

Entre los razonadores disponibles, se detallan aquellos que cumplen con las características y requisitos del proyecto, en términos de expresividad, algoritmo de inferencia, lenguaje de

implementación, velocidad de inferencia y compatibilidad con lenguajes de definición de ontologías, reglas y herramientas para la implementación de ontologías.

Existen muchos razonadores que han ido surgiendo a lo largo de los años. Por ello, se han dividido en generaciones según el año en el que se publicaron.

Bossam es un razonador que permite razonar sobre ontologías OWL/OWL2 y soporta reglas SWRL. Además, dispone de varias interfaces de acceso como por la línea de comandos o por API Java, lo que facilita la integración de reglas con Java. Pellet es un razonador basado en lógica descriptiva que permite verificar la consistencia de las ontologías OWL. Soporta SPARQL, reglas SWRL y se puede acceder a él por medio de editores de ontologías, como Protégé. Pellet permite el razonamiento incremental, es decir, la capacidad de procesar la ontología sin necesidad de realizar todos los pasos previos desde cero. Esto es importante para la gestión en NRT/RT de las amenazas.

Existen otros razonadores como HermiT y FaCT/FaCT++ que también permiten razonar con ontologías OWL y soportan el acceso a través de Protégé. Todos se detallan en los apartados siguientes.

#### 3.3.4.1 BOSSAM

Bossam [26] es un razonador semántico usado en la Web Semántica, basado en el algoritmo RETE, un algoritmo de reconocimiento de patrones usado para inferir conocimiento a partir de una base de hechos, pero que hace que aumente el tiempo de ejecución.

Bossam utiliza un algoritmo de razonamiento deductivo o, dicho de otra manera, de cadena dirigida por los hechos, encadenamiento hacia delante o *forward chaining*. Los razonadores que se basan en los algoritmos de razonamiento deductivo parten de los datos disponibles en base a los hechos, y a partir de ellos extraen más datos mediante reglas de inferencia hasta que se alcanza un objetivo. Se aplican las reglas para obtener como consecuencia nuevos resultados.

Bossam permite razonar sobre ontologías basadas en lenguajes OWL y RDF/RDFS y soporta reglas definidas en RuleML y SWRL. Además, para acceder al motor de inferencia de Bossam existen interfaces de acceso, como la línea de comandos, APIs en Java, una interfaz web o a través de editores de ontologías como Protégé a través de *plugins*.

Cabe destacar que Bossam facilita la integración de reglas con Java y se centra en ser de fácil acceso web y proporcionar razonamiento distribuido. Sin embargo, no optimiza el consumo de recursos.

#### 3.3.4.2 PELLET

Pellet [27] es un motor de razonamiento semántico OWL DL de código abierto basado en Java. Está basado en la lógica descriptiva y soporta servicios avanzados de razonamiento sobre ontologías OWL. Permite validar y comprobar la consistencia de las ontologías, hacer clasificaciones de clases y responder a consultas SPARQL.

El razonador Pellet realiza inferencia de conocimiento haciendo uso de los algoritmos de razonamiento hacia delante y hacia atrás (*forward* y *backward chaining*). Además, Pellet permite realizar debuggeado de ontologías detectando conflictos y realizando diagnóstico y resolución de errores. De esta manera, se comprueba la correcta consistencia de la ontología.

Pellet permite razonar sobre ontologías definidas en OWL-DL y OWL 2 y soporta reglas SWRL, implementando un *parser* de reglas en SWRL, de manera que se puede pasar de DL-safe rules (reglas demasiado básicas) a reglas SWRL. Pellet soporta SPARQL/SPARQL-DL (en nuevas versiones de Pellet) para realizar consultas sobre los datos representados en la ontología. Además,

es multiplataforma y soporta clasificación incremental, es decir, que el proceso de inferencia no se realiza desde el principio si ya se ha realizado el proceso de clasificación anteriormente.

Además, para acceder a Pellet existen diversas interfaces de acceso como APIs en Java (OWLAPI o Jena), por línea de comandos o *plugins* en editores de ontologías como Protégé o SWOOP y mediante el protocolo DIG.

Por otro lado, se ha demostrado mediante varios estudios [28] [29] que Pellet es el razonador más rápido en el proceso de inferencia de conocimiento, el de menor tiempo de respuesta y soporta un gran número de características, por lo que ha sido definido dentro del grupo de razonadores sólidos.

Existen muchas aplicaciones que hacen uso de Pellet y está en continuo desarrollo apareciendo cada día nuevas versiones.

#### 3.3.4.3 FACT/FACT++

*Fast Classification of Terminologies* (FaCT) [30] es un razonador semántico basado en lógica descriptiva para verificar la consistencia de los modelos lógicos, haciendo uso del algoritmo de cálculo *tableaux*. FaCT está escrito en Common LISP (lenguaje usado en inteligencia artificial) y se ha ejecutado con éxito en entornos comerciales, como GNU.

FaCT se caracteriza por su lógica expresiva que le permite razonar con bases de datos, su optimización en forma de tablas y su estructura cliente-servidor basada en CORBA (arquitectura estándar para sistemas de objetos distribuidos).

FaCT++ es un razonador de código abierto que extiende las características y técnicas de optimización de FaCT, pero con una arquitectura interna diferente e implementado en C++. Soporta los lenguajes de ontologías OWL DL y OWL 2 DL con algunas restricciones y se puede hacer uso de FaCT++ a través de editores como Protégé, mediante el protocolo DIG.

Cabe destacar que tiene un bajo tiempo de respuesta, pero no soporta el lenguaje de reglas SWRL.

#### 3.3.4.4 HERMIT

HermiT [31] es el primer razonador semántico disponible públicamente que proporciona una alta eficiencia de razonamiento haciendo uso del algoritmo de cálculo *hypertableau*. HermiT está implementado en Java y soporta ontologías OWL 2, consultas SPARQL y reglas SWRL.

HermiT es un razonador semántico OWL, usa semántica directa y pasa todas las pruebas de conformidad del OWL 2 para los razonadores de semántica directa. HermiT es el primer razonador que proporciona un rápido procesamiento que reduce enormemente el número de pruebas de consistencia necesarias para calcular las jerarquías de clase y de propiedad para clasificar ontologías complejas.

Se puede hacer uso de este razonador utilizando editores de ontologías como Protégé, en el que se dispone de un *plugin* para HermiT. También se puede acceder a través de línea de comandos o API Java.

HermiT proporciona buen rendimiento en ontologías e incluye soporte para algunas características de ontologías no estándar, como gráficos de descripción. Sin embargo, tiene un tiempo de respuesta alto y no es más rápido que otros razonadores, como Pellet.

VENTAJAS	DESVENTAJAS
<b>Bossam</b>	
<ul style="list-style-type: none"> <li>○ Compatible con OWL, RDF/RDFS, RuleML y SWRL.</li> <li>○ Interfaz de acceso API Java o web.</li> <li>○ Plugin en Protégé.</li> <li>○ Integración de reglas en Java.</li> </ul>	<ul style="list-style-type: none"> <li>○ No optimiza el uso de recursos.</li> <li>○ Software comercial</li> </ul>
<b>Pellet</b>	
<ul style="list-style-type: none"> <li>○ Compatible con OWL DL, OWL2, SWRL y consultas SPARQL.</li> <li>○ Interfaz de acceso API Java.</li> <li>○ Plugin Protégé.</li> <li>○ Rápida inferencia de conocimiento.</li> <li>○ Bajo tiempo de respuesta.</li> <li>○ Código abierto.</li> </ul>	<ul style="list-style-type: none"> <li>○ Necesidad de mejorar su compatibilidad completa con SWRL.</li> </ul>
<b>FaCT/FaCT++</b>	
<ul style="list-style-type: none"> <li>○ Código abierto.</li> <li>○ Compatible con IWL DL y OWL DL 2.</li> <li>○ Interfaz de acceso a través del protocolo DIG en Protégé.</li> <li>○ Bajo tiempo de respuesta.</li> </ul>	<ul style="list-style-type: none"> <li>○ No soporta SWRL</li> </ul>
<b>HermiT</b>	
<ul style="list-style-type: none"> <li>○ Código abierto.</li> <li>○ Compatible con OWL, OWL2, SPARQL y SWRL.</li> <li>○ Plugin en Protégé.</li> <li>○ Interfaz de acceso API Java.</li> <li>○ Rápida clasificación.</li> </ul>	<ul style="list-style-type: none"> <li>○ Alto tiempo de respuesta.</li> </ul>

Tabla 4. Comparación de los razonadores semánticos.

## 4 DISEÑO GENERAL DEL SISTEMA

---

En este apartado se describe el diseño general del sistema y se especifica la funcionalidad global del mismo. Partiendo de los requisitos funcionales y no funcionales y los requisitos de entrada y salida, se establece la arquitectura general más conveniente para cumplir los objetivos del sistema. Previamente a la definición de los requisitos se justifica el uso de ontologías en el sistema y se seleccionan las tecnologías y herramientas adecuadas para su desarrollo.

Como ya se ha especificado con anterioridad, el sistema propuesto está basado en las ontologías. El uso de ontologías permite la inserción de información de fuentes heterogéneas en un sistema. Cada una de estas fuentes tiene un tipo de sintaxis y formatos distintos. Por lo que si la base del sistema es una ontología se puede integrar esa información en una estructura común de clases, propiedades y relaciones y una semántica formal. Aunque cada información sea independiente de las demás, dentro de la ontología se pueden relacionar fuentes diversas entre sí de modo que la información tenga sentido semántico. De esta manera, puede ser procesada por un razonador semántico para inferir nuevo conocimiento.

Se propone el uso del lenguaje OWL para la definición de la ontología. Utilizar el lenguaje OWL ofrece la posibilidad utilizar herramientas para el procesamiento y razonamiento de la información definida en la ontología. Además, como se menciona en la Tabla 2, OWL es el lenguaje más utilizado en la Web Semántica y permite utilizar distintas construcciones para la definición de la información del sistema propuesto, que está basada en datos sobre anomalías, amenazas y riesgos.

Puesto que la semántica del lenguaje de ontologías es formal, establecer relaciones entre los datos puede hacerse de manera automática puesto que estos serán legibles e interpretables por máquinas a alta velocidad. Esto conlleva un beneficio sobre la detección de amenazas y riesgos, ya que cuanto antes se detecten, más rápida será la respuesta.

Para definir las relaciones y el comportamiento que debe tener el sistema una vez procesada esa información, se necesitan reglas. En este caso, se utilizan reglas en formato SWRL. Como se menciona en la Tabla 3, el lenguaje SWRL extiende las restricciones que se definen con el lenguaje OWL para determinar de manera más precisa el comportamiento del sistema y de los elementos que se integran en el mismo y es el recomendado por la Web Semántica.

Por otro lado, se hará uso del razonamiento semántico. Las ontologías incluyen una gran cantidad de clases y propiedades, lo que hace necesario que exista un motor que se encarga de verificar su correcta construcción. Además, los razonadores también generarán nuevo conocimiento a partir de las relaciones que se establecen dentro de la ontología y las reglas. Como se menciona en la Tabla 4, el razonador que más se adecúa al sistema es el razonador Pellet, puesto que es el que ofrece una respuesta más rápida y es compatible con el lenguaje de ontologías OWL y el lenguaje SWRL de reglas que se han seleccionado.

Por tanto, el objetivo de este trabajo es implementar un sistema que sea capaz de detectar la llegada de datos relacionados con anomalías, *Threat Intelligence* y activos. Estos datos proceden de fuentes externas independientes y tienen una sintaxis distinta. Por ello, deben introducirse en el sistema mediante una definición unificada para crear una estructura común. Como resultado se generan amenazas y riesgos. Los riesgos son analizados, evaluados y gestionados y se estima el riesgo resultante dinámicamente.

A continuación, se especifica el ámbito de proyecto al que pertenece este trabajo, que marcan en gran medida los requisitos y la arquitectura general del sistema que se explica posteriormente.

## 4.1 ÁMBITO DEL PROYECTO

Este trabajo se enmarca en un proyecto más amplio que se encuentra en su fase de diseño y desarrollo. El proyecto tiene como objetivo diseñar, desarrollar y validar un prototipo de Plataforma Avanzada de Conciencia Cibersituacional. Mediante esta plataforma se pretende monitorizar fuentes de información heterogéneas, procesar los datos recogidos mediante técnicas de aprendizaje automático y sistemas expertos basados en ontologías y reglas de especificación formal comportamiento.

La finalidad del proyecto es que sea aplicable en un entorno organizativo para detectar y predecir patrones avanzados de ataques, así como para estimar el riesgo de exposición a los mismos.

Puesto que se trata de un proyecto extenso, el sistema completo se divide en varios módulos. La arquitectura general de este sistema se puede observar en la siguiente figura.

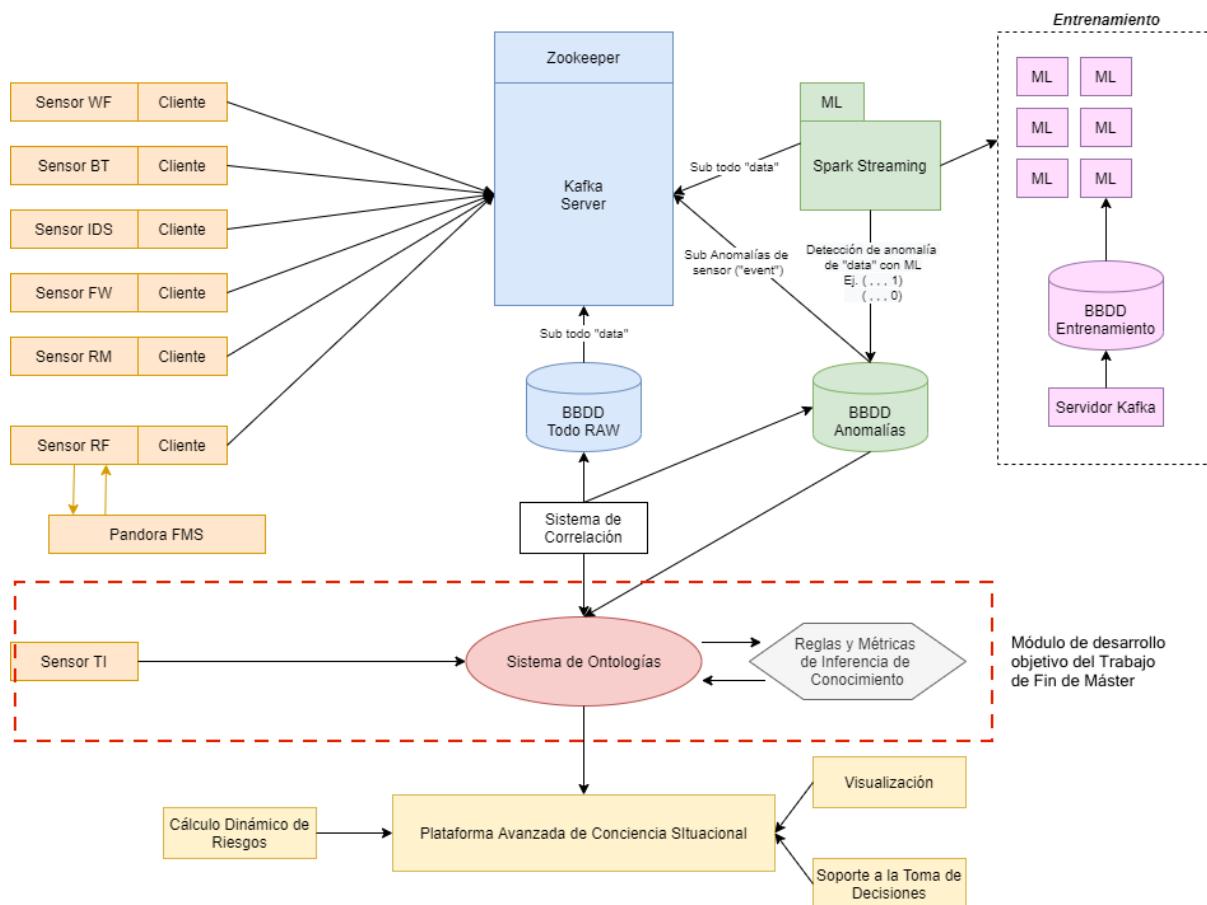


Figura 4. Arquitectura global del proyecto en el que se enmarca el TFM

Como se observa en la figura, el sistema de ontologías es el módulo al que pertenece este trabajo. Se trata de utilizar sistemas expertos basados en ontologías para realizar razonamientos que permitan implementar métricas de seguridad que realimentan y perfeccionan el propio modelo.

El desarrollo del proyecto tiene como principal novedad la inclusión de fuentes heterogéneas como pueden ser sensores, indicadores de presencia, fuentes de inteligencia de amenazas, etc., cuya correlación sirve para enriquecer los sistemas de conciencia situacional hacia un nivel superior, para que transmitan de manera más realista lo que está sucediendo.

## 4.2 REQUISITOS

Antes de especificar el diseño de la arquitectura del sistema, se necesita determinar previamente los requisitos que debe satisfacer el sistema para poder cumplir los objetivos. Se realiza un análisis de los requisitos de cada una de las partes del sistema. Se identifican:

- Requisitos funcionales y no funcionales.
- Requisitos de entrada.
- Requisitos de salida.

Esta identificación se utiliza para describir cuáles son las funcionalidades que debe cumplir el sistema y se especifican a nivel de sistema global.

- |            |   |
|------------|---|
| <b>R01</b> | El sistema deberá detectar la llegada de nuevos datos y actualizarlos si es necesario.                      |
| <b>R02</b> | El sistema deberá integrar los datos en una estructura común basada en ontologías.                          |
| <b>R03</b> | El sistema deberá identificar amenazas y riesgos.   |
| <b>R04</b> | El sistema deberá analizar los riesgos en potenciales y residuales.   |
| <b>R05</b> | El sistema deberá proporcionar soluciones para gestionar los riesgos.                                       |
| <b>R06</b> | El sistema deberá calcular el riesgo total al que está expuesto la organización.                            |
| <b>R07</b> | El sistema deberá considerar el entorno y el historial de resultados de riesgos para el cálculo del riesgo. |
| <b>R08</b> | El sistema deberá almacenar los resultados obtenidos.   |
| <b>R09</b> | El sistema deberá ser lo más rápido posible.  |

Tabla 5. Requisitos funcionales y no funcionales del sistema.

Los requisitos R01 y R02 son fundamentales para poder introducir información nueva y actualizar el sistema con dicha información. Tener actualizada la información del sistema es imprescindible para datos como los referentes a las anomalías, ya que puede que una anomalía de manera individualizada no suponga una amenaza, pero la ocurrencia de otras con características similares sea indicio de una posible amenaza.

Los requisitos R03, R04 y R05, son imprescindibles para realizar la funcionalidad propia del análisis de riesgos, de manera similar a como lo hacen metodologías como MAGERIT, pero de manera automática. El resto de los requisitos R06 y R07 son necesarios para el cálculo del riesgo. De esta manera, el cálculo del riesgo debe de considerar los históricos y se realizará dinámicamente y en tiempo real. El requisito R08 es importante para que se pueda disponer de los resultados obtenidos por el sistema en el futuro y para el cálculo del riesgo. Sin duda el sistema debe cumplir el requisito R09 puesto que un sistema que no procese rápido la información no podrá ejecutar acciones en tiempo real o quasi real.

- |             |   |
|-------------|---|
| <b>RE01</b> | El sistema deberá adaptar la información de entrada de distintas estructuras sintácticas. |
|-------------|---|

Tabla 6. Requisito de entrada del sistema

El requisito de entrada que se identifica es fundamental para introducir los datos de manera efectiva en el sistema. Además, debe de poder adaptarse para poder recibir información de otras fuentes aún no consideradas y así escalar.

- |             |  |
|-------------|--|
| <b>RS01</b> | El sistema deberá poder acceder al historial de los resultados del análisis y cálculo del riesgo.  |
| <b>RS02</b> | El sistema deberá poder generar gráficos relevantes sobre los resultados. Estos gráficos deben ser entendibles por personas no expertas y analistas. |
| <b>RS03</b> | El sistema deberá proponer estrategias de tratamiento de los riesgos en función de su severidad.   |

**Tabla 7. Requisitos de salida del sistema**

En cuanto a los requisitos de salida, el requisito RS01 es fundamental para que el sistema pueda acceder a los datos almacenados previamente. Estos datos son imprescindibles para poder representar los resultados en gráficos, según lo dispuesto en el requisito RS02. Además, mediante la representación de los resultados se facilitan las tareas a nivel estratégico proporcionando sugerencias de cómo gestionar los riesgos detectados: monitorizándolos, mitigándolos, etc.

### 4.3 CASOS DE USO

En este apartado se definen los casos de uso del sistema que serán validados posteriormente. Estos casos de uso muestran las funcionalidades del sistema y son detallados a bajo nivel en diagramas de secuencia que se encuentran debajo de cada caso de uso.

<b>CU01</b> DETECCIÓN Y ACTUALIZACIÓN DE ANOMALÍAS.	
<b>DEPENDENCIAS</b>	<p>R01- El sistema deberá detectar la llegada de nuevos datos y actualizarlos si es necesario.</p> <p>R02 - El sistema deberá integrar los datos en una estructura común basada en ontologías.</p> <p>RE01 - El sistema deberá adaptar la información de entrada de distintas estructuras sintácticas.</p>
<b>PRECONDICIÓN</b>	El sistema dispone de un fichero JSON donde la información de las anomalías de fuentes externas se almacena y se actualiza.
<b>DESCRIPCIÓN</b>	Este caso de uso detecta la existencia de nueva información relativa a anomalías procedente de una fuente de datos externa. Si ya existen anomalías con información similar se relacionan y se actualizan.
<b>SECUENCIA</b>	<p>a. Detección de una actualización debido a información relacionada con anomalías.</p> <p>b. Se registra la anomalía en la ontología y se actualiza la información.</p>
<b>POSTCONDICIÓN</b>	El sistema integra las anomalías en la ontología y la actualiza.

**Tabla 8. Caso de uso CU01: Detección y actualización de anomalías**

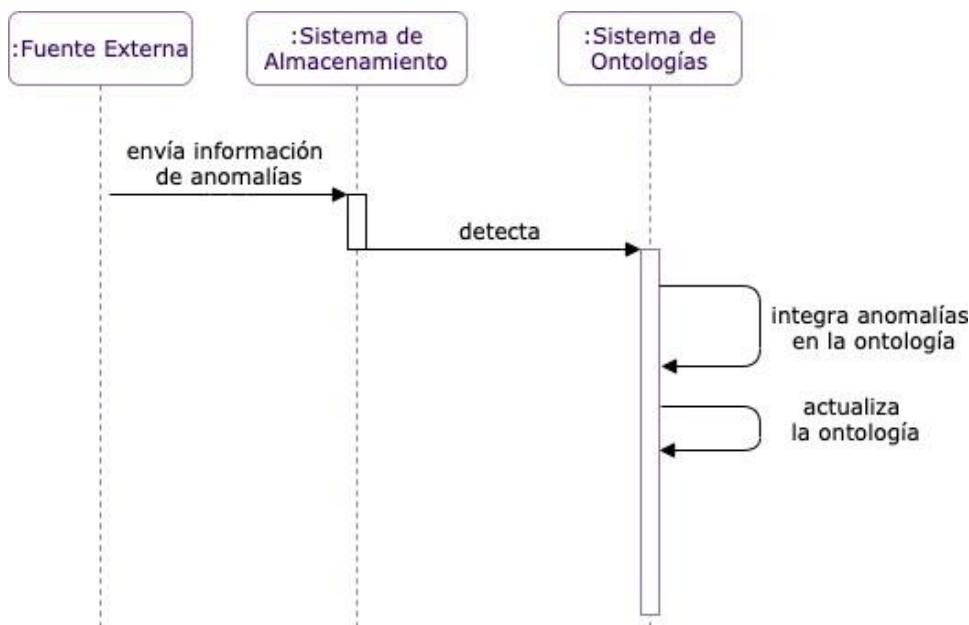


Figura 5. Diagrama de secuencia del caso de uso CU01

CU02 DETECCIÓN Y ACTUALIZACIÓN DE INFORMACIÓN DE <i>THREAT INTELLIGENCE</i> .	
<b>DEPENDENCIAS</b>	R01- El sistema deberá detectar la llegada de nuevos datos y actualizarlos si es necesario. R02 - El sistema deberá integrar los datos en una estructura común basada en ontologías. RE01 - El sistema deberá adaptar la información de entrada de distintas estructuras sintácticas.
<b>PRECONDICIÓN</b>	El sistema dispone de un fichero JSON donde la información de TI se almacena y se actualiza.
<b>DESCRIPCIÓN</b>	Este caso de uso detecta la existencia de nueva información de TI procedente de un sensor.
<b>SECUENCIA</b>	a. Detección de una actualización. b. Se registra la información de TI en la ontología.
<b>POSTCONDICIÓN</b>	El sistema detecta la existencia de la nueva información y la integra en la ontología.

Tabla 9. Caso de uso CU02: Detección y actualización de información de Threat Intelligence

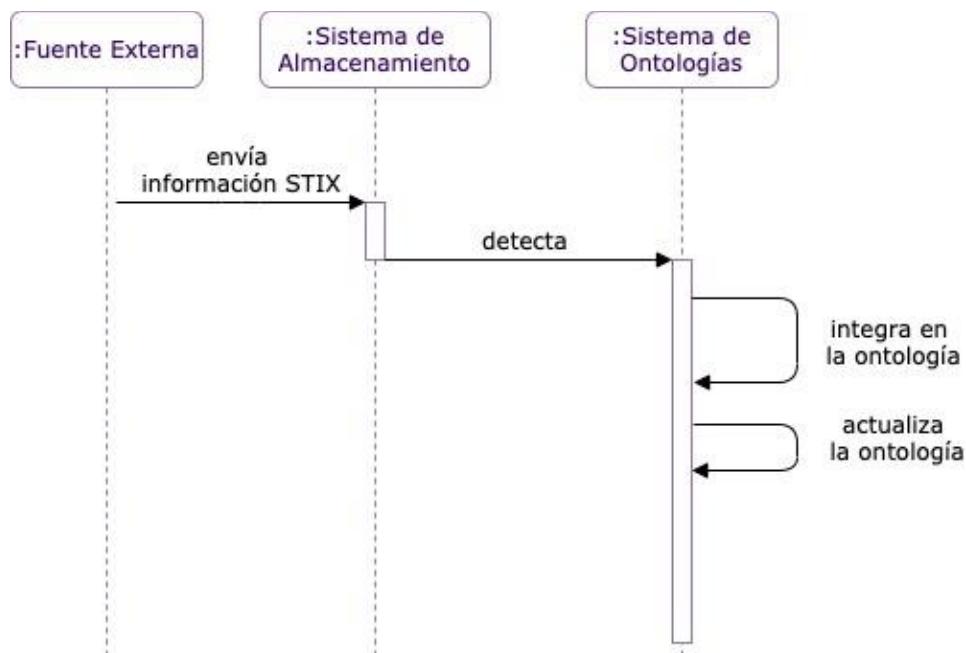


Figura 6. Diagrama de secuencia del caso de uso CU02

### CU03

#### INTRODUCCIÓN DE DATOS DE LOS ACTIVOS.

<b>DEPENDENCIAS</b>	R02 - El sistema deberá integrar los datos en una estructura común basada en ontologías. RE01 - El sistema deberá adaptar la información de entrada de distintas estructuras sintácticas.
<b>PRECONDICIÓN</b>	El sistema dispone de un fichero CSV que tiene la información de salida de la herramienta PILAR y que almacena información sobre los activos y sus dependencias. El sistema también dispone de otro fichero en formato XML con las valoraciones de dichos activos.
<b>DESCRIPCIÓN</b>	Este caso de uso integra en la ontología la información relacionada con los activos de la organización, así como su valoración.
<b>SECUENCIA</b>	a. Ejecución del sistema. b. Se integra la información de los activos. c. Se registra la valoración de los activos.
<b>POSTCONDICIÓN</b>	El sistema dispone de los activos y sus valoraciones.

Tabla 10. Caso de uso CU03: Introducción de datos de activos

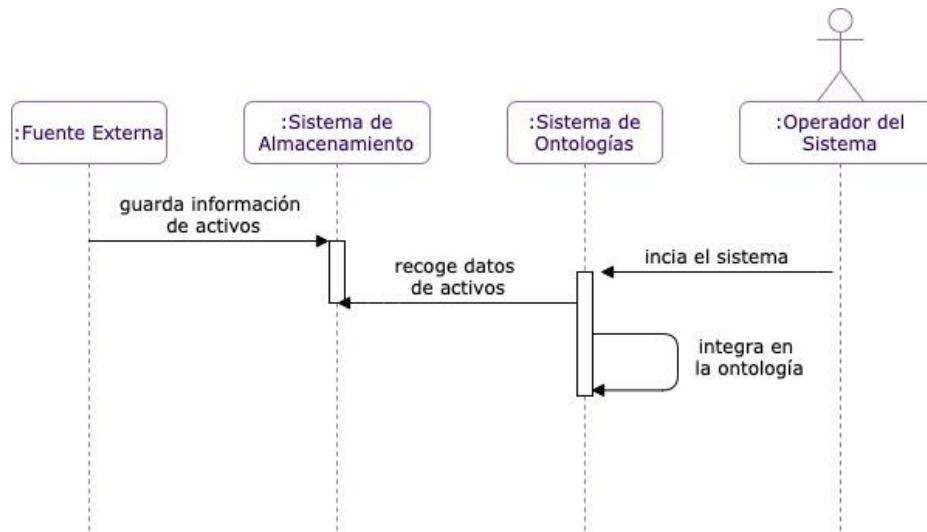


Figura 7. Diagrama de secuencia del caso de uso CU03

CU04 RAZONAMIENTO SEMÁNTICO DE LA ONTOLOGÍA.	
<b>DEPENDENCIAS</b>	<p>R03 - El sistema deberá identificar amenazas y riesgos.</p> <p>R04 - El sistema deberá analizar los riesgos en potenciales y residuales.</p> <p>R05 - El sistema deberá proporcionar soluciones para gestionar los riesgos.</p> <p>R06 - El sistema deberá calcular el riesgo total al que está expuesto la organización.</p> <p>R07 - El sistema deberá considerar el entorno y el historial de resultados de riesgos para el cálculo del riesgo.</p> <p>R08 - El sistema deberá almacenar los resultados obtenidos.</p> <p>R09 - El sistema deberá ser lo más rápido posible.</p> <p>RS01 - El sistema deberá poder acceder al historial de los resultados del análisis y cálculo del riesgo.</p> <p>RS03 - El sistema deberá proponer estrategias de tratamiento de los riesgos en función de su severidad.</p>
<b>PRECONDICIÓN</b>	Debe existir una ontología con información suficiente e instancias y unas reglas de inferencia en formato SWRL.
<b>DESCRIPCIÓN</b>	Este caso de uso se realiza de manera automática una vez se actualiza la ontología, escrita en lenguaje OWL. La ontología se actualiza con nuevas instancias y se ejecutan las reglas para deducir nuevo conocimiento. La ontología pasa a un nuevo estado y se calcula el riesgo dinámico.
<b>SECUENCIA</b>	<ol style="list-style-type: none"> <li>Actualización de la ontología.</li> <li>Se procesa la información de la ontología en base a las reglas.</li> <li>Se verifica la consistencia de la ontología.</li> <li>Se calcula el riesgo en función de los datos actualizados.</li> </ol>

<b>POSTCONDICIÓN</b>	El sistema almacena resultados de riesgo dinámico en un historial.
----------------------	--

Tabla 11. Caso de uso CU04: Razonamiento semántico de la ontología

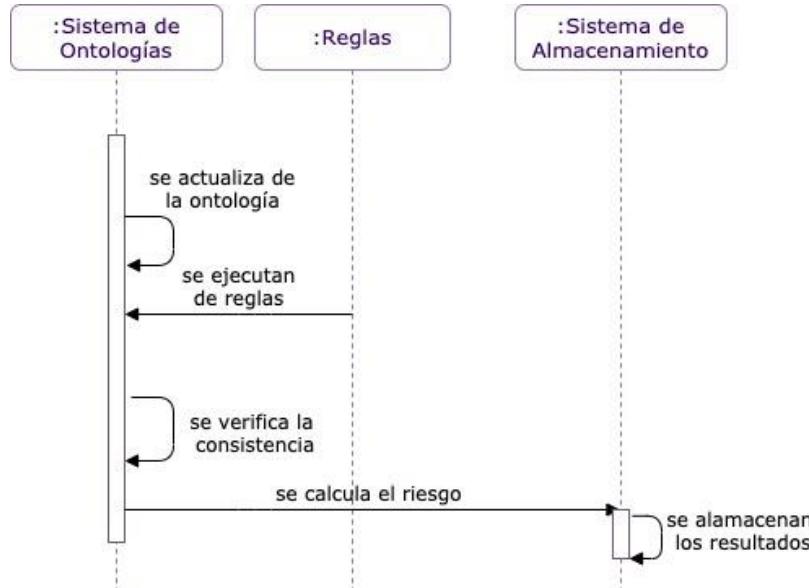


Figura 8. Diagrama de secuencia del caso de uso CU04

CU05		VISUALIZACIÓN DE RESULTADOS
<b>DEPENDENCIAS</b>	RS02- El sistema deberá poder generar gráficos relevantes sobre los resultados. Estos gráficos deben ser entendibles por personas no expertas y analistas. RS03 - El sistema deberá proponer estrategias de tratamiento de los riesgos en función de su severidad.	
<b>PRECONDICIÓN</b>	El sistema dispone de un fichero JSON donde se almacenan los resultados del cálculo del riesgo y del soporte a la toma de decisiones.	
<b>DESCRIPCIÓN</b>	Este caso de uso muestra los resultados del riesgo de la organización en diferentes niveles (a nivel global y de manera individual para cada riesgo).	
<b>SECUENCIA</b>	a. El analista o persona no experta pulsa el botón deseado. b. Se registra el botón pulsado y se solicitan los datos al sistema de almacenamiento. c. Se envía los datos. d. La información es mostrada a la persona que lo ha solicitado.	
<b>POSTCONDICIÓN</b>	El sistema ofrece una interfaz para la visualización de los resultados de manera gráfica.	

Tabla 12. Caso de uso CU05: Visualización de resultados

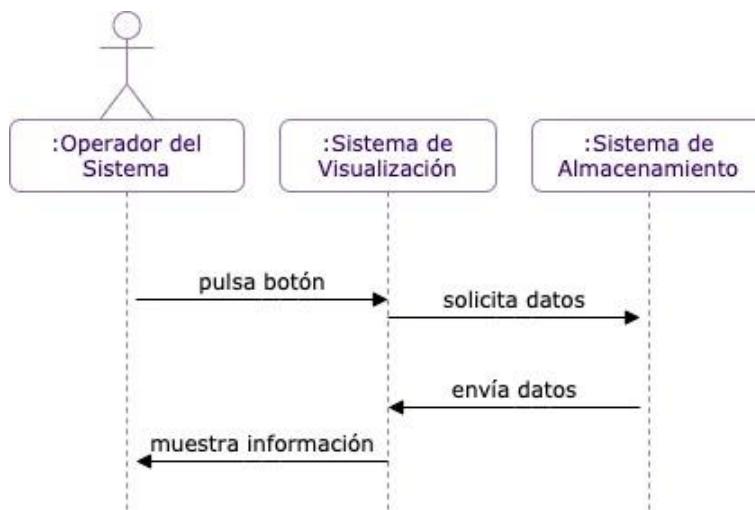


Figura 9. Diagrama de secuencia del caso de uso CU05

#### 4.4 ARQUITECTURA DEL SISTEMA

Una vez se han identificado los requisitos y casos de uso del sistema, se lleva a cabo el diseño de la arquitectura del sistema.

La arquitectura del sistema propuesta, que se puede observar en la Figura 10. El módulo principal del sistema es el sistema de ontologías. Como resultado se genera el cálculo del riesgo dinámico. Ya se ha expuesto con anterioridad la necesidad de realizar un análisis dinámico del riesgo debido a la rapidez con la que pueden cambiar las amenazas. También es importante debido a la llegada de datos relevantes de inteligencia de amenazas, procedentes de otros organismos asociados mediante el formato STIX.

El resultado del cálculo del riesgo se visualiza gráficamente mediante una interfaz. Además, los riesgos obtenidos pueden clasificarse de manera que se puedan sacar de esta clasificación conclusiones sobre las estrategias más adecuadas conforme al valor del riesgo calculado. Esta información también se observará mediante una interfaz gráfica.

Esta manera de definir el sistema permite generar datos e información necesarios en los distintos niveles de una organización: estratégico, táctico y operacional.

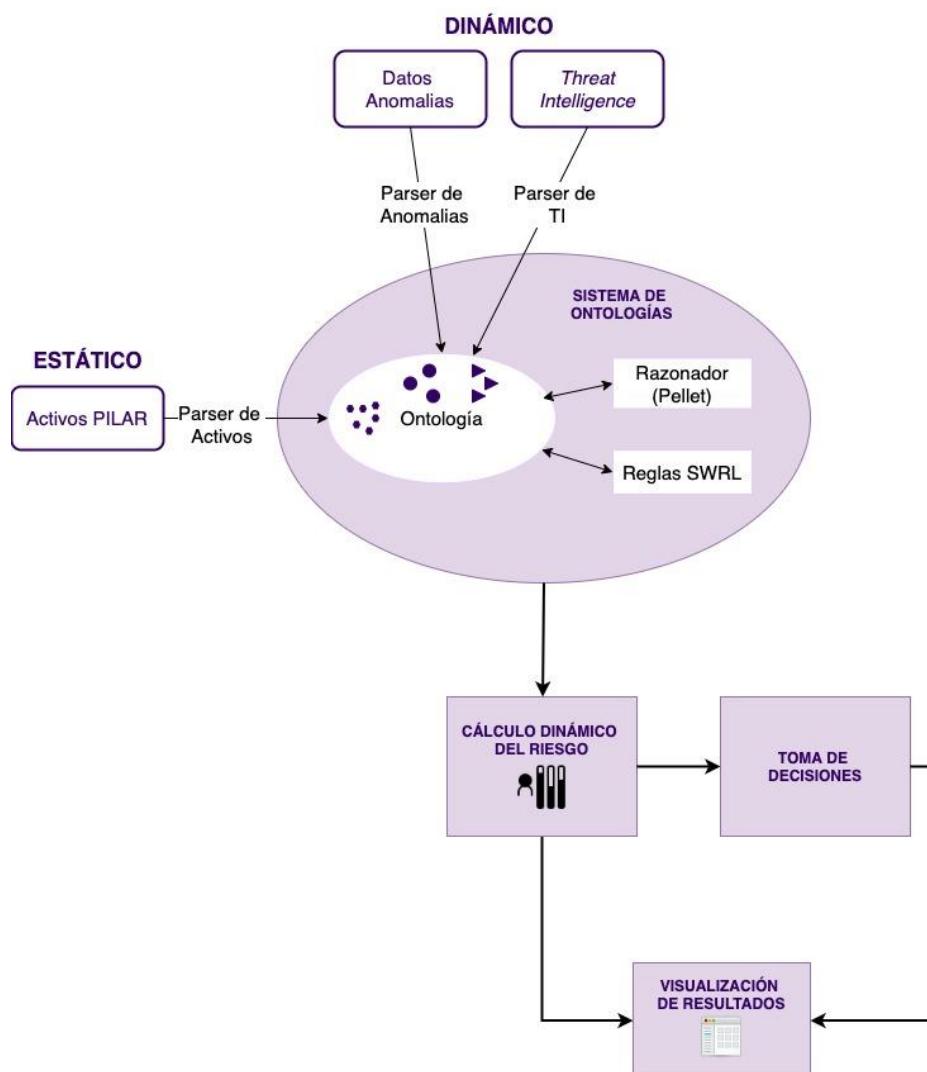


Figura 10. Arquitectura general del sistema.

El sistema se divide en cuatro subsistemas según su función: subsistema de ontologías, subsistema del cálculo dinámico del riesgo, subsistema de visualización y subsistema de soporte a la toma de decisiones.

#### ○ Subsistema de Ontologías

El sistema principal es el subsistema de ontologías. Se encarga de introducir los datos que llegan de fuentes externas dinámicas y estáticas: datos de los activos, bases de datos de anomalías e información de *Threat Intelligence*. El sistema actualiza y representa dichos datos en una estructura unificada según las clases, propiedades y relaciones que se han definido en la ontología.

Sobre este subsistema se desarrollan las reglas en formato SWRL para definir las métricas de seguridad del sistema y se realiza el razonamiento sobre la ontología, a fin de verificar su consistencia y de inferir nueva información sobre el dominio de conocimiento.

Los datos de salida de este subsistema son las amenazas y riesgos que se generan como resultado del proceso de inferencia sobre la ontología y sirve como datos de entrada para el subsistema del cálculo del riesgo y para el subsistema de soporte a la toma de decisiones.

Cada una de las partes de este subsistema se describen en el capítulo Diseño de la Ontología.

#### ○ Subsistema del Cálculo del Riesgo

El subsistema del cálculo del riesgo recoge las amenazas y riesgos procedentes del subsistema de ontologías y se encarga de realizar el proceso de evaluación de riesgos. El subsistema evalúa los riesgos en potenciales y residuales. A partir de esta evaluación el subsistema calcula los riesgos potenciales y residuales asociados a cada uno de los riesgos generados.

En función de los resultados obtenidos se calcula el riesgo potencial total y el riesgo residual total de la organización cada instante, así como los riesgos potenciales y residuales totales teniendo en cuenta las medidas del riesgo en el pasado. Esto se hace así porque se considera que el cálculo del riesgo no es un cálculo discreto, sino que se ve afectado por los riesgos que ya se han producido.

Como resultado, se almacenan los datos calculados en formato JSON para poder disponer de ellos en los siguientes cálculos y sirven como datos de entrada para el subsistema de visualización.

La funcionalidad de este subsistema se detalla en el apartado de Cálculo Dinámico del Riesgo.

#### ○ Subsistema de Soporte a la Toma de Decisiones

El subsistema de soporte a la toma de decisiones recoge los riesgos generados por el subsistema de ontologías y realiza una clasificación de los riesgos según su gravedad. Divide los riesgos en Gravedad Extrema, Gravedad Alta, Gravedad Media y Gravedad Baja. Según esta clasificación se proponen estrategias para el tratamiento de cada uno de los riesgos, como estrategias de mitigación, estrategias de monitorización, estrategias de investigación el riesgo o estrategias de respuesta a un incidente.

Para facilitar la visualización de las estrategias propuestas, se ha añadido en el subsistema de visualización un mapa de riesgos. En este mapa se dibujan los riesgos en función de su impacto y probabilidad y se proponen procedimientos a seguir en función de la estrategia propuesta para cada riesgo.

Los datos de salida del subsistema de soporte para la toma de decisiones es el conjunto de los riesgos clasificados en función de su gravedad, las estrategias propuestas para cada riesgo generado en el subsistema de cálculo del riesgo.

La funcionalidad de este subsistema se implementa mediante las reglas SWRL y una interfaz gráfica. Por tanto, cada una de sus partes se detalla en los apartados correspondientes a los mismos.

#### ○ Subsistema de Visualización

El subsistema de visualización recoge los datos procedentes del subsistema de cálculo del riesgo y del subsistema soporte para la toma de decisiones. Este subsistema se encarga de generar una interfaz para observar gráficamente los resultados obtenidos.

La interfaz se realiza mediante la librería JFreeChart de Java y facilita la comprensión de los datos procedentes de los subsistemas anteriores. Por un lado, se pueden observar los riesgos potenciales y residuales finales, así como la evolución de sus valores en función de medidas realizadas anteriormente.

Por otro lado, se pueden observar los valores de los riesgos potenciales y residuales de cada riesgo. Además, para apoyar a la toma de decisiones, se genera un mapa de riesgos con el fin de

---

evaluar los riesgos obtenidos, decidir qué riesgos se van a tratar y observar las tareas propuestas de actuación.

## 5 DISEÑO DE LA ONTOLOGÍA

Este apartado tiene como objetivo la descripción detallada de la ontología propuesta y las métricas de seguridad. Se propone el uso de estándares con el fin de que la implementación y el desarrollo conjunto con otras organizaciones sea sencillo en el futuro.

Este trabajo pretende ser una consolidación de distintas partes sobre un sistema de ontologías para la gestión y tratamiento de anomalías, amenazas y riesgos. Por ello, uno de los subsistemas principales es el subsistema de ontologías.

La ventaja de utilizar las ontologías radica en su capacidad de proporcionar una representación que une las características de distintas partes y las relaciona. Además, a partir de esa representación se puede deducir conocimiento automáticamente y con un sentido semántico. Esto permite que puedan utilizarse máquinas que entiendan dicho lenguaje para realizar procesamientos mucho más rápidos que si se hiciera de manera manual.

A pesar de que las ontologías se han desarrollado en mayor medida en el ámbito de la medicina, utilizarlas para el entorno de la ciberseguridad conlleva muchos beneficios. Y es que, como se ha mencionado anteriormente, son muchos los elementos de seguridad que existen, y los expertos insisten en no decantarse por uno en concreto, sino utilizar un conjunto de todos ellos. Este conjunto de elementos se complementa y proporciona una visión más amplia del sistema y de las amenazas y riesgos a los que se enfrenta.

Sin embargo, sino se analizan adecuadamente los datos que proporcionan los distintos elementos de seguridad, no servirán e, incluso, generarán más incertidumbre. Por eso, se propone el uso de ontologías en este trabajo. Gracias a ellas se soluciona el problema de la heterogeneidad semántica de los datos procedentes de fuentes diversas. Los datos se mapean a un modelo integrado semánticamente, se clasifican en clases y se establecen relaciones entre ellos. Así se asegura que la información sea procesada adecuadamente por el sistema.

Como se ha visto en el apartado de *Ontologías: Modelos formales de representación de la información* son muchos los lenguajes existentes para la definición de ontologías, así como los lenguajes de definición de reglas y razonadores semánticos. A pesar de que suele ser complicado encontrar ontologías existentes que cubran las necesidades de un sistema concreto, el presente proyecto se basa en el trabajo realizado por Raúl Riesco en su Tesis Doctoral [32]. En esta tesis, el autor desarrolla una ontología que permite procesos de gestión y evaluación de riesgo más dinámicos que los actuales, en tiempo real o casi en tiempo real. El trabajo que aquí nos ocupa, se basa en esta ontología y añade nuevas clases que permiten la incorporación de información relacionada con anomalías. De la misma manera, se toma como referencia las definiciones de las métricas de seguridad y se añaden nuevas para definir el comportamiento debido a incorporación de los nuevos datos sobre anomalías.

En los apartados siguientes se detallan las contribuciones del presente proyecto para adaptar la ontología base con el fin de cumplir los objetivos del sistema.

### 5.1 DEFINICIÓN DE LA ONTOLOGÍA

En primer lugar, es importante especificar qué se quiere representar con la ontología y para qué se va a usar. El dominio que se desea modelar es el relacionado con toda aquella información de anomalías, amenazas y riesgos que pueda ser extraída de datos recogidos a través de fuentes externas diversas.

El proceso sería el siguiente: se detecta la existencia de nueva información y se integra en el sistema. Este será el encargado de procesarla lo más rápido posible para generar amenazas y evaluar y gestionar el riesgo de una organización.

La tesis de Raúl Riesco [32] define un marco de Gestión y Evaluación Dinámica de Riesgos (DRM/ DRA). Para ello, implementa una ontología híbrida de riesgos (DRM) y amenazas (CTI), basada en el lenguaje OWL y el formato STIX.

La ontología, que se define en este trabajo, recoge esa ontología híbrida y la adapta añadiendo una nueva para integrar datos sobre anomalías. La ontología final será aquella formada por: una ontología que recoge los datos STIX (ontología CTI), una ontología del ámbito de riesgos (ontología DRM) y una ontología que integra los datos de anomalías (ontología ONA). Cabe destacar que otra contribución del presente trabajo ha sido actualizar la ontología CTI a la nueva versión 2.1 expuesta por OASIS para los elementos STIX [33], en la que las definiciones de algunos elementos, sus atributos y relaciones han sido modificados.

En la siguiente figura se puede observar como quedan relacionadas entre sí cada una de las ontologías que conforman la ontología principal.

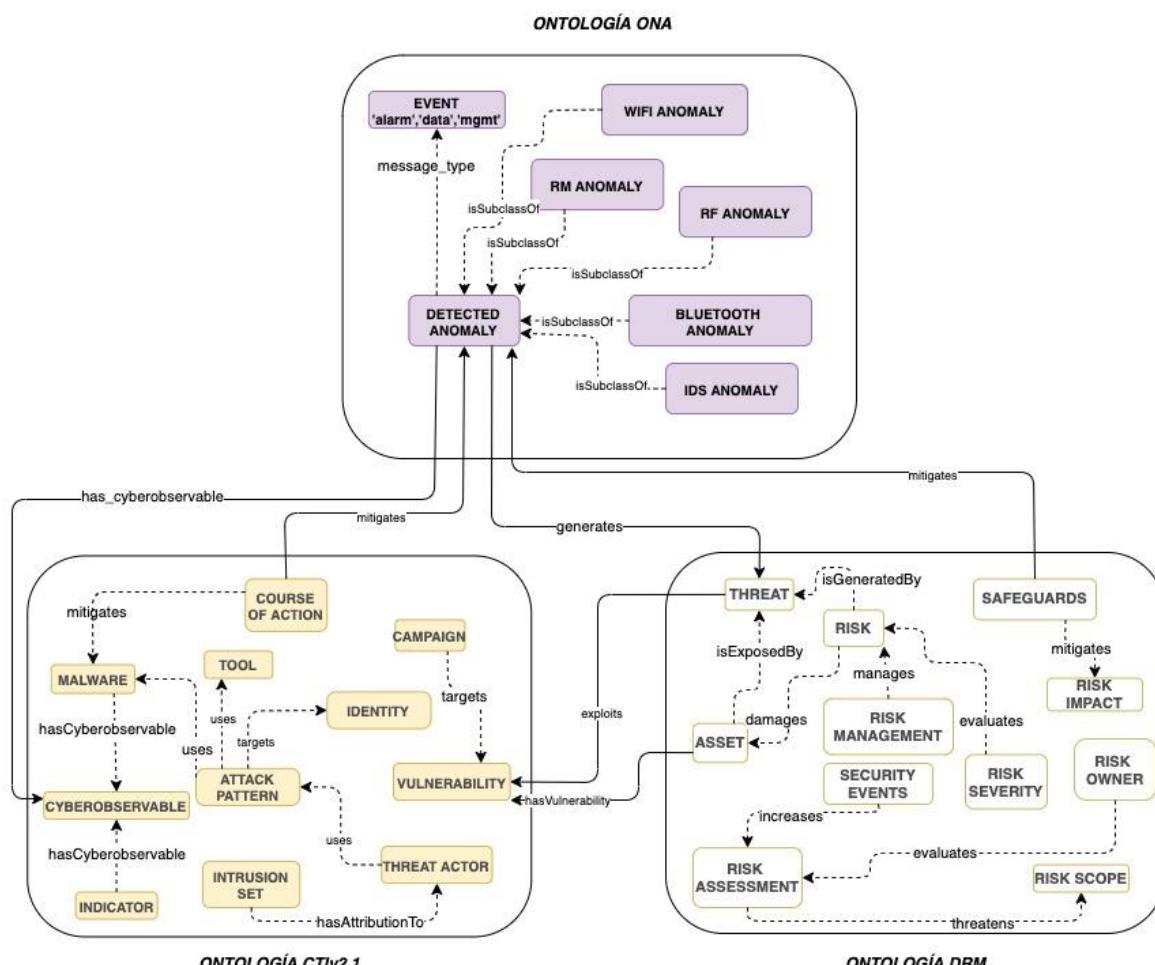


Figura 11. Relaciones entre las diferentes ontologías.

### 5.1.1 CLASES DE LA ONTOLOGÍA

Tras el análisis realizado en el apartado Lenguajes de Definición de Ontologías, la Tabla 2 refleja que los lenguajes más expresivos y utilizados son OWL y OWL2. Es por ello, por lo que para el desarrollo de la ontología se utilizan estos lenguajes.

La estructura de ambos lenguajes es muy similar, pero es OWL2 el que ofrece mayor expresividad, permite un mayor número de axiomas, restricciones y propiedades de cardinalidad. Esto se debe a que OWL2 extiende la semántica y sintaxis de OWL. Además, es compatible con la herramienta Protégé, editor utilizado para el desarrollo de la ontología, con el formato de reglas SWRL y con muchos razonadores semánticos, como Pellet.

La ontología definida en el trabajo representa las anomalías detectadas por fuentes de distintos tipos:

- Anomalía de tipo Wifi.
- Anomalía de tipo Bluetooth.
- Anomalía de tipo Radiofrecuencia.
- Anomalía de tipo Redes Móviles.
- Anomalía de tipo Ciberseguridad (ciberseguridad hace referencia a todo tipo de herramientas de seguridad que puedan dar información sobre el sistema, como IDS, SIEM, etc.)

La definición y especificación de la ontología ONA incluye relaciones con las ontologías mencionadas anteriormente como se observa en la Figura 11. El dominio de esta ontología incluye todos los procesos de gestión y análisis relacionados con anomalías. La ontología propuesta aprovecha las relaciones con la ontología DRM y CTI para enlazar la información de anomalías con el ámbito de la ciberseguridad.

A continuación, se detalla cada una de las clases de la ontología implementada, sus propiedades, sus relaciones y características.

#### 5.1.1.1 CLASE ANOMALÍA DETECTADA

NOMBRE ANOMALÍA DETECTADA	
DESCRIPCIÓN	Identifica las anomalías procedentes de fuentes externas.
PROPIEDADES	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"detected-anomaly"^^xsd:string}</li> <li>○ <b>start_time</b> some xsd:dateTimeStamp</li> <li>○ <b>end_time</b> some xsd:dateTimeStamp</li> <li>○ <b>id</b> some xsd:string</li> <li>○ <b>suspicious_value</b> some xsd:float</li> </ul>
RELACIONES CON OTRAS CLASES	<ul style="list-style-type: none"> <li>○ <b>message_type</b> some Event</li> <li>○ <b>isMitigatedBy</b> some Course_of_Action</li> <li>○ <b>isMitigatedBy</b> some Safeguard</li> <li>○ <b>caused_by</b> some Asset</li> <li>○ <b>has_cyberobservable</b> some Cyber_Observable</li> <li>○ <b>generates</b> some Threat</li> </ul>
SUBCLASES	<ul style="list-style-type: none"> <li>○ WiFi_Sensor_Anomaly</li> <li>○ Bluetooth_Sensor_Anomaly</li> <li>○ Cybersecurity_Sensor_Anomaly</li> <li>○ RF_Sensor_Anomaly</li> <li>○ RM_Sensor_Anomaly</li> <li>○ UBA_Sensor_Anomaly</li> <li>○ Physical_Presence_Sensor_Anomaly</li> </ul>

Tabla 13. Definición de la ontología. Clase Anomalía Detectada

Las características de las propiedades se describen a continuación:

- **type (obligatorio)**  
Tipo: *string*.  
Definición: *valor del campo debe ser “detected-anomaly”*.
- **start\_time (opcional)**  
Tipo: *timestamp*.  
Definición: *fecha y hora en las que se vio por primera vez la anomalía*.
- **end\_time (opcional)**  
Tipo: *timestamp*.  
Definición: *fecha y hora en las que se vio por última vez la anomalía*.
- **id (obligatorio)**  
Tipo: *string*.  
Definición: *identifica a la anomalía detectada*.
- **suspicious\_value (obligatorio)**  
Tipo: *float*.  
Definición: *representa el valor que hace que la anomalía sea capaz por sí misma de generar una amenaza que conlleve un riesgo*.
- **caused\_by some Asset (opcional)**  
Tipo: *Asset*.  
Definición: *hace referencia al activo sospechoso de ser la fuente de la anomalía y/o su causa. Esto ocurrirá principalmente si la anomalía se produce a partir de un componente registrado como activo en la organización*.
- **has\_cyberobservable some Cyber\_Observable (opcional)**  
Tipo: *Cyber\_Observable*.  
Definición: *hace referencia a los ciberobservables que se han podido observar en el dominio de ciberseguridad debido a la ocurrencia de la anomalía*.
- **message\_type some Event (opcional)**  
Tipo: *Event*.  
Definición: *representa el tipo de eventos que pueden enviar las fuentes externas: ALARM, DATA o MGMT*.
- **generates some Threat (opcional)**  
Tipo: *Threat*.  
Definición: *representa la amenaza que genera la anomalía*.

#### 5.1.1.2 CLASE ANOMALÍA DE SENSOR WIFI

NOMBRE		ANOMALÍA DE SENSOR WIFI
DESCRIPCIÓN	Información de anomalía tipo Wifi.	
PROPIEDADES	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"WF"^^xsd:string}</li> <li>○ <b>start_time</b> some xsd:dateTimeStamp</li> <li>○ <b>end_time</b> some xsd:dateTimeStamp</li> <li>○ <b>id</b> some xsd:string</li> <li>○ <b>suspicious_value</b> some xsd:float</li> <li>○ <b>has_essid</b> some xsd:string</li> <li>○ <b>pwr</b> some xsd:float</li> <li>○ <b>apwr</b> some xsd:float</li> </ul>	

	<ul style="list-style-type: none"> <li>○ <b>clients</b> some xsd:integer</li> <li>○ <b>visits</b> some xsd:integer</li> </ul>
<b>RELACIONES CON OTRAS CLASES</b>	<ul style="list-style-type: none"> <li>○ <b>message_type</b> some Event</li> <li>○ <b>isMitigatedBy</b> some Course_of_Action</li> <li>○ <b>isMitigatedBy</b> some Safeguard</li> <li>○ <b>caused_by</b> some Asset</li> <li>○ <b>has_cyberobservable</b> some Cyber_Observable</li> <li>○ <b>generates</b> some Threat</li> <li>○ <b>related-to</b> some Mac-Addr</li> <li>○ <b>(src_ref</b> some IPV4ADDR) or (<b>src_ref</b> some IPV6ADDR)</li> </ul>
<b>SUBCLASES</b>	No tiene subclases.

Tabla 14. Definición de la ontología. Clase Anomalía Wifi

Las propiedades que caracterizan la clase de anomalía WiFi de las demás, a parte de las ya comentadas en el apartado de anterior, se describen a continuación:

- **has\_essid**  
Tipo: *string*.  
Definición: *identifica el SSID del punto de acceso al que estaba conectado el cliente.*
- **pwr**  
Tipo: *float*.  
Definición: *identifica la potencia medida con la que se ha medido el dispositivo en el intervalo de medición.*
- **apwr**  
Tipo: *float*.  
Definición: *identifica la potencia media del punto de acceso visto por el sensor.*
- **clients**  
Tipo: *integer*.  
Definición: *identifica el número de clientes vistos durante la medición.*
- **visits**  
Tipo: *integer*.  
Definición: *identifica el número de veces que el cliente ha sido detectado en el intervalo de medición (entradas y reentradas).*
- **related-to**  
Tipo: *MAC-Addr*.  
Definición: *esta característica esta identificada por el campo ‘userid’ procedente de los datos de las fuentes externas. Sirve para identificar la dirección MAC del dispositivo del usuario que ha sido detectado.*
- **src\_ref**  
Tipo: *IPV4ADDR o IPV6ADDR*.  
Definición: *identifica la dirección IP, si existe, del dispositivo que ha sido detectado.*

### 5.1.1.3 CLASE ANOMALÍA DE SENSOR BLUETOOTH

NOMBRE		ANOMALÍA DE SENSOR BLUETOOTH
DESCRIPCIÓN	Información de anomalía tipo Bluetooth.	
PROPIEDADES	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"BT"^^xsd:string}</li> <li>○ <b>start_time</b> some xsd:dateTimeStamp</li> <li>○ <b>end_time</b> some xsd:dateTimeStamp</li> <li>○ <b>id</b> some xsd:string</li> <li>○ <b>suspicious_value</b> some xsd:float</li> <li>○ <b>name</b> some xsd:string</li> <li>○ <b>has_RSSI</b> some xsd:float</li> </ul>	
RELACIONES CON OTRAS CLASES	<ul style="list-style-type: none"> <li>○ <b>message_type</b> some Event</li> <li>○ <b>isMitigatedBy</b> some Course_of_Action</li> <li>○ <b>isMitigatedBy</b> some Safeguard</li> <li>○ <b>caused_by</b> some Asset</li> <li>○ <b>has_cyberobservable</b> some Cyber_Observable</li> <li>○ <b>generates</b> some Threat</li> <li>○ <b>related-to</b> some Mac-Addr</li> </ul>	
SUBCLASES	No tiene subclases.	

Tabla 15. Definición de la ontología. Clase Anomalía Bluetooth

Las propiedades que caracterizan la clase Anomalía Bluetooth de los demás, a parte de las ya comentadas en el apartado de la clase Anomalía Detectada, se describen a continuación:

- **name**  
Tipo: *string*.  
Definición: *identifica el nombre del dispositivo*.
- **has\_RSSI**  
Tipo: *float*.  
Definición: *identifica la potencia con la que se recibe la señal identificada con el campo 'rss' de los datos de las fuentes externas*.
- **related-to**  
Tipo: *MAC-Addr*.  
Definición: *identifica la dirección MAC del dispositivo detectado, identificada por el campo 'address' procedente de los datos de las fuentes externas*.

### 5.1.1.4 CLASE ANOMALÍA DE SENSOR DE CIBERSEGURIDAD

NOMBRE		ANOMALÍA DE SENSOR DE CIBERSEGURIDAD
DESCRIPCIÓN	Información de anomalía tipo Ciberseguridad.	
PROPIEDADES	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"IDS"^^xsd:string}</li> <li>○ <b>start_time</b> some xsd:dateTimeStamp</li> <li>○ <b>end_time</b> some xsd:dateTimeStamp</li> <li>○ <b>id</b> some xsd:string</li> </ul>	

	<ul style="list-style-type: none"> <li><input type="radio"/> <b>suspicious_value</b> some xsd:float</li> <li><input type="radio"/> <b>prediction</b> some xsd:string</li> <li><input type="radio"/> <b>srcip</b> some xsd:string</li> <li><input type="radio"/> <b>sport</b> some xsd:string</li> <li><input type="radio"/> <b>dstip</b> some xsd:string</li> <li><input type="radio"/> <b>dsport</b> some xsd:string</li> <li><input type="radio"/> <b>proto</b> some xsd:string</li> </ul>
<b>RELACIONES CON OTRAS CLASES</b>	<ul style="list-style-type: none"> <li><input type="radio"/> <b>message_type</b> some Event</li> <li><input type="radio"/> <b>isMitigatedBy</b> some Course_of_Action</li> <li><input type="radio"/> <b>isMitigatedBy</b> some Safeguard</li> <li><input type="radio"/> <b>caused_by</b> some Asset</li> <li><input type="radio"/> <b>has_cyberobservable</b> some Cyber_Observable</li> <li><input type="radio"/> <b>generates</b> some Threat</li> </ul>
<b>SUBCLASES</b>	No tiene subclases.

Tabla 16. Definición de la ontología. Clase Anomalía Ciberseguridad

Las propiedades que caracterizan la clase Anomalía de Ciberseguridad de los demás, a parte de las ya comentadas en el apartado de la clase Anomalía Detectada, se describen a continuación:

- prediction**  
Tipo: *string*.  
Definición: *identifica que la información es una anomalía*.
- srcip**  
Tipo: *string*.  
Definición: *identifica la dirección IP fuente*.
- sport**  
Tipo: *string*.  
Definición: *identifica el puerto de la fuente*.
- dstip**  
Tipo: *string*.  
Definición: *identifica la dirección IP del destino*.
- dsport**  
Tipo: *string*.  
Definición: *identifica el puerto destino*.
- proto**  
Tipo: *string*.  
Definición: *identifica el protocolo utilizado*.

#### 5.1.1.5 CLASE ANOMALÍA DE SENSOR DE RADIOFRECUENCIA

NOMBRE	ANOMALÍA DE SENSOR DE RADIOFRECUENCIA
<b>DESCRIPCIÓN</b>	Información de anomalía tipo Radiofrecuencia.
<b>PROPIEDADES</b>	<ul style="list-style-type: none"> <li><input type="radio"/> <b>type</b> some {"RF"^^xsd:string}</li> <li><input type="radio"/> <b>start_time</b> some xsd:dateTimeStamp</li> <li><input type="radio"/> <b>end_time</b> some xsd:dateTimeStamp</li> <li><input type="radio"/> <b>id</b> some xsd:string</li> </ul>

	<ul style="list-style-type: none"> <li>○ <b>suspicious_value</b> some xsd:float</li> <li>○ <b>has_modulation</b> some xsd:string</li> <li>○ <b>has_signal_frecuency</b> some xsd:string</li> <li>○ <b>has_signal_power</b> some xsd:string</li> <li>○ <b>payload_bin</b> some xsd:hexBinary</li> </ul>
<b>RELACIONES CON OTRAS CLASES</b>	<ul style="list-style-type: none"> <li>○ <b>message_type</b> some Event</li> <li>○ <b>isMitigatedBy</b> some Course_of_Action</li> <li>○ <b>isMitigatedBy</b> some Safeguard</li> <li>○ <b>caused_by</b> some Asset</li> <li>○ <b>has_cyberobservable</b> some Cyber_Observable</li> <li>○ <b>generates</b> some Threat</li> </ul>
<b>SUBCLASES</b>	No tiene subclases.

Tabla 17. Definición de la ontología. Clase Anomalía Radiofrecuencia

Las propiedades que caracterizan la clase Anomalía de Radiofrecuencia de las demás, a parte de las ya comentadas en el apartado de clase Anomalía Detectada, se describen a continuación:

- **has\_modulation**  
Tipo: *string*.  
Definición: *representa la modulación de la señal que es detectada: OOK, 2FSK o NONE, identificada por el campo 'mod' en los datos procedentes de las fuentes externas.*
- **has\_signal\_frecuency**  
Tipo: *string*.  
Definición: *representa la frecuencia de la señal que es detectada, en MHz, identificada por el campo 'freq' en los datos procedentes de las fuentes externas.*
- **has\_signal\_power**  
Tipo: *string*.  
Definición: *representa el nivel de intensidad de la señal recibida en dBms, identificado por el campo 'signal' en los datos procedentes de las fuentes externas.*
- **payload\_bin**  
Tipo: *hexBinary*.  
Definición: *representa los datos asociados o extraídos de la señal, identificados por el campo 'payload' en los datos procedentes de las fuentes externas.*

#### 5.1.1.6 CLASE ANOMALÍA DE SENSOR DE REDES MÓVILES

NOMBRE	ANOMALÍA DE SENSOR DE REDES MÓVILES
<b>DESCRIPCIÓN</b>	Información de anomalía tipo Redes Móviles.
<b>PROPIEDADES</b>	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"RM"^^xsd:string}</li> <li>○ <b>start_time</b> some xsd:dateTimeStamp</li> <li>○ <b>end_time</b> some xsd:dateTimeStamp</li> <li>○ <b>id</b> some xsd:string</li> <li>○ <b>suspicious_value</b> some xsd:float</li> <li>○ <b>has_IMEI</b> some xsd:string</li> <li>○ <b>has_IMSI</b> some xsd:string</li> <li>○ <b>rat</b> some xsd:string</li> </ul>

RELACIONES CON OTRAS CLASES	<ul style="list-style-type: none"> <li>○ <b>message_type</b> some Event</li> <li>○ <b>isMitigatedBy</b> some Course_of_Action</li> <li>○ <b>isMitigatedBy</b> some Safeguard</li> <li>○ <b>caused_by</b> some Asset</li> <li>○ <b>has_cyberobservable</b> some Cyber_Observable</li> <li>○ <b>generates</b> some Threat</li> </ul>
SUBCLASES	No tiene subclases.

Tabla 18. Definición de la ontología. Clase Anomalía Redes Móviles

Las propiedades que caracterizan la clase de anomalía de Redes Móviles de las demás, a parte de las ya comentadas en el apartado de clase Anomalía Detectada, se describen a continuación:

- **has\_IMEI**  
Tipo: *string*.  
Definición: *representa el dispositivo móvil detectado, identificado por el campo ‘imei’ en los datos procedentes de las fuentes externas.*
- **has\_IMSI**  
Tipo: *string*.  
Definición: *representa el valor del identificador internacional del abonado en el dispositivo móvil detectado, identificado en el campo ‘imsi’ en los datos procedentes de las fuentes externas.*
- **rat**  
Tipo: *string*.  
Definición: *representa el tipo de acceso radio: 2G, 3G y 4G, identificado en el campo ‘rat’ en los datos procedentes de las fuentes externas.*

#### 5.1.1.7 EVENTO

NOMBRE	EVENTO
DESCRIPCIÓN	Evento por el que se ha identificado la anomalía. Puede ser ‘ALARM’ (evento de alarma), ‘MGMT’ (evento de gestión) o ‘DATA’ (evento de recogida de datos).
PROPIEDADES	<ul style="list-style-type: none"> <li>○ <b>type</b> some {"event"^^xsd:string}</li> <li>○ <b>value</b> only xsd:string</li> </ul>
RELACIONES CON OTRAS CLASES	No tiene relaciones con otras clases.
SUBCLASES	No tiene subclases.

Tabla 19. Definición de la ontología. Clase Evento

Las características de las propiedades se describen a continuación:

- **type (obligatorio)**  
Tipo: *string*.  
Definición: *valor del campo debe ser “event”*.

- **value (opcional)**

Tipo: *string*.

Definición: *identifica el evento por el que se ha producido la anomalía. Esto es ‘ALARM’, ‘MGMT’ o ‘DATA’.*

## 5.2 MÉTRICAS DE SEGURIDAD

Para que el sistema pueda analizar y procesar cada una de las clases y atributos que conforman nuestra ontología, se necesita hacer uso de métricas de seguridad. Se ha especificado anteriormente que los lenguajes de definición de ontologías, tales como OWL y OWL2, no son capaces de expresar restricciones complejas. Es por ello, por lo que se introducen las reglas SWRL y el razonador semántico.

En la ontología definida existen muchos parámetros susceptibles de ser medidos y controlados. En función de ellos se puede establecer el comportamiento del sistema. Esos parámetros se incluyen en las reglas para inferir nuevo conocimiento sobre amenazas y riesgos. La precisión de los resultados dependerá de la definición de dichas reglas.

El uso de las reglas y el razonador semántico permite introducir información nueva en el sistema de manera controlada. El razonador verifica la consistencia de los datos, tanto de los que se integran nuevamente como de los que se infieren a partir de las reglas. Gracias a la inferencia de conocimiento se pueden medir parámetros como el impacto, la probabilidad, el número de amenazas, el riesgo potencial o el riesgo residual.

En los siguientes apartados se definen las reglas en formato SWRL.

### 5.2.1 REGLAS DE COMPORTAMIENTO

Para la definición de las reglas, como ya se ha indicado anteriormente, se utiliza el lenguaje SWRL. SWRL es uno de los lenguajes más utilizados para la definición de reglas y se ha elegido por las siguientes razones:

- **Compatibilidad:** SWRL es compatible con los lenguajes de definición de ontologías OWL y OWL2, así como con la mayor parte de razonadores y editores de ontologías.
- **Expresividad:** permite representar el comportamiento con una expresividad mayor que otros lenguajes, como los comparados en la Tabla 3, y además expresa mayores restricciones que los axiomas de OWL2.
- **Fácil integración:** presenta una fácil integración con editores de ontologías (Protégé) y razonadores semánticos (Pellet) y una facilidad de uso debido a su amplia adopción y su constante evolución.

Una característica común que tienen SWRL y OWL es la monotonía, que significa que las reglas no pueden utilizarse para modificar información existente en la ontología. Los razonadores OWL actuales solo implementan reglas DL-Safe SWRL. Esto significa que SWRL se basa en OWL DL, que proporciona más expresividad que OWL DL solo. Las reglas DL-Safe SWRL son un subconjunto de las reglas SWRL y restringen el poder expresivo de SWRL. El hecho de que se utilice este tipo de reglas significa que se restringen las reglas para garantizar que estas trabajen solo con individuos conocidos de la ontología. Puesto que en este trabajo se utiliza el razonador Pellet, las reglas que utiliza serán reglas DL-Safe SWRL.

Por ello, se implementa un programa en Java, que será detallado más adelante, con el fin de tratar la información para que cumpla con los objetivos deseados y superar las restricciones que imponen estos lenguajes.

Al igual que ocurre en la definición de la ontología, las reglas que establece Raúl Riesco en su tesis [32] son adaptadas al sistema y se añaden otras nuevas para tener en cuenta los datos sobre anomalías. Las reglas que define Raúl Riesco en su tesis y a las que se presta especial atención se clasifican en los siguientes tipos:

- Reglas de Valoración de Activos.
- Reglas de Inventario de Amenazas.
- Reglas de Inventario de Riesgos.
- Reglas de Evaluación de Riesgos.
- Reglas de Gravedad de Riesgos.
- Reglas de Políticas de Seguridad.
- Reglas de Gestión de Riesgos.
- Reglas de Inteligencia de Amenazas.

Las reglas SWRL también pueden hacer uso de *built-ins* que sirven para extender la expresividad de las reglas SWRL. Un *built-in* es un predicado que toma uno o más argumentos y se evalúa como verdadero si los argumentos satisfacen dicho predicado. Por ejemplo, se puede definir un *built-in lessThan* que acepta dos argumentos y devuelve verdadero si se cumple que el primer argumento es menor que el segundo.

Gracias a estos *built-ins* la expresividad de SWRL aumenta. En las reglas mencionadas anteriormente se utilizan varios tipos. Se identifican utilizando el atributo *swrlx: built-in*. Cabe destacar la existencia del *built-in swrlx:makeOWLThing*, mediante el cual se añade la funcionalidad de poder crear individuos a partir de las reglas.

Como se ha mencionado anteriormente, el sistema propuesto ofrece un soporte a la toma de decisiones. Para ello, se ha hecho uso de las ‘Reglas de Gravedad de Riesgos’ y de las ‘Reglas de Gestión de Riesgos’ por los que se clasifican los riesgos obtenidos según su gravedad y se proponen estrategias de actuación en función del valor final del riesgo.

Tomando como base la definición de los tipos de reglas mencionados, se generan otras reglas para definir el comportamiento en función de los datos de anomalías. Además, se crean otro tipo nuevo de reglas específico para las anomalías y al que se ha llamado ‘Reglas de Anomalías’. Las reglas que se han definido para este trabajo se detallan a continuación:

### **Reglas de Anomalías: Anomalía Wifi**

Las anomalías que se integran en la ontología tienen un parámetro en común, llamado *suspicious value*. Este valor representa la probabilidad de que una anomalía suponga una amenaza para la organización. Teniendo en cuenta esto se han supuesto tres escenarios:

- a. Una anomalía con un parámetro *suspicious\_value* de valor inferior a un determinado umbral no genera ninguna amenaza por sí misma.
- b. Una anomalía con un parámetro *suspicious\_value* superior a un umbral supone una amenaza por sí misma y, por tanto, se genera una amenaza.
- c. Una anomalía junto a la existencia de una amenaza aumenta un riesgo.

El umbral es un parámetro que define el valor para que una anomalía sea considerada amenaza o no. Existe otro parámetro denominado intervalo, que es el valor con el que aumenta el parámetro *suspicious\_value*. Ambos valores son configurados por el usuario del sistema a través de un archivo de configuración.

En el caso de anomalías Wifi, la característica común que hace que el valor del parámetro *suspicious\_value* aumente es el valor de la MAC del dispositivo del que proceden. Si ese valor es igual para varias anomalías, puede significar que existe alguien que usa un dispositivo deliberadamente para tener acceso no autorizado a la organización.

Esta forma de proceder puede afectar la dimensión de confidencialidad, pudiendo darse fugas o filtraciones de información. Por ello, el valor de *suspicious\_value* aumenta según el intervalo y cuando supera el umbral se genera automáticamente la amenaza con la siguiente regla.

REGLA	ANOMALIES WIFI
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Deliberated Unauthorized Access</i> debido a la existencia de varias anomalías wifi, procedentes de un mismo dispositivo. Se crean amenazas si el valor del parámetro <i>suspicious value</i> supera un umbral.
FÓRMULA	$\text{WiFi\_Sensor\_Anomaly(?w)} \wedge \text{suspicious\_value(?w, ?s)} \wedge \text{swrlb:greaterThanOrEqual(?s, umbral)} \wedge \text{swrlx:makeOWLThing(?x, ?w)}$ $\rightarrow \text{probability(?x, 2.0^^xsd:float)} \wedge \text{DeliberatedUnauthorizedAccess(?x)}$ $\wedge (\text{?x, "Threat Deliberated Unauthorized Access"}) \wedge \text{impact(?x, "4.0"^^xsd:float)} \wedge \text{isGeneratedBy(?x, ?w)} \wedge \text{numType(?x, 14)}$

Tabla 20. Reglas de Anomalías: Anomalía Wifi

### Reglas de Anomalías: Anomalía Bluetooth

En el caso de bluetooth ocurre lo mismo que en el caso de una anomalía Wifi. Si existe una concurrencia de anomalías que proceden del mismo dispositivo, la probabilidad de que esa anomalía suponga una amenaza aumenta y el valor del parámetro *suspicious\_value* aumentará también. Cuando supere el umbral, se genera automáticamente una amenaza con la siguiente regla.

REGLA	ANOMALIES BLUETOOTH
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Deliberated Unauthorized Access</i> debido a la existencia de varias anomalías bluetooth, procedentes de un mismo dispositivo. Se crean amenazas si el valor del parámetro <i>suspicious value</i> supera un umbral.
FÓRMULA	$\text{Bluetooth\_Sensor\_Anomaly(?w)} \wedge \text{suspicious\_value(?w, ?s)} \wedge \text{swrlb:greaterThanOrEqual(?s, umbral)} \wedge \text{swrlx:makeOWLThing(?x, ?w)}$ $\rightarrow \text{probability(?x, "2.0"^^xsd:float)} \wedge \text{DeliberatedUnauthorizedAccess(?x)} \wedge \text{type(?x, "Threat Deliberated Unauthorized Access")}$ $\wedge \text{impact(?x, "4.0"^^xsd:float)} \wedge \text{isGeneratedBy(?x, ?w)} \wedge \text{numType(?x, 14)}$

Tabla 21. Reglas de Anomalías: Anomalía Bluetooth

### Reglas de Anomalías: Anomalía Radiofrecuencia

En el caso de la existencia de una anomalía de radiofrecuencia, la probabilidad de que exista una amenaza aumenta si existen varias anomalías con la misma frecuencia de señal. Esto puede significar que existen dispositivos que intenta colapsar el espectro. Esto puede suponer ataques de denegación de servicio y por tanto va en contra de la disponibilidad.

Por ello, cuando el parámetro *suspicious\_value* supera el umbral, se genera una amenaza de tipo *Denial of Service*. Si se produce la Denegación de Servicio, la disponibilidad de los servicios de la organización se ve afectada, teniendo consecuencias directas e indirectas sobre la organización y sus servicios. Esto se genera automáticamente con la siguiente regla.

REGLA	ANOMALIES RADIOFRECUENCIA
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Denial of Service</i> debido a la existencia de varias anomalías de tipo radiofrecuencia, con la misma frecuencia. Se crean amenazas si el valor del parámetro <i>suspicious value</i> supera un umbral.
FÓRMULA	$\text{RF_Sensor_Anomaly(?w) \wedge suspicious_value(?w, ?s) \wedge swrlb:greaterThanOrEqual(?s, umbral) \wedge swrlx:makeOWLThing(?x, ?w) -> probability(?x, "2.0"^^xsd:float) \wedge DenialOfService(?x) \wedge type(?x, "Threat Denial Of Service") \wedge impact(?x, "4.0"^^xsd:float) \wedge isGeneratedBy(?x, ?w) \wedge numType(?x, 15)}$

Tabla 22. Reglas de Anomalías: Anomalía Radiofrecuencia

### Reglas de Anomalías: Anomalía Redes Móviles

En este caso, la probabilidad de que exista una amenaza aumentará si varias anomalías de redes móviles coinciden en su valor de IMEI. Esto significa que las fuentes de datos tienen información de un mismo dispositivo móvil que utiliza la red móvil y que si genera varias anomalías puede suponer una amenaza para la organización. Esta anomalía genera automáticamente una amenaza de tipo *Deliberated Unauthorized Access* mediante la siguiente regla.

REGLA	ANOMALIES REDES MÓVILES
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Deliberated Unauthorized Access</i> debido a la existencia de varias anomalías de tipo redes móviles, procedentes de un mismo dispositivo. Se crean amenazas si el valor del parámetro <i>suspicious value</i> supera un umbral.
FÓRMULA	$\text{RM_Sensor_Anomaly(?w) \wedge suspicious_value(?w, ?s) \wedge swrlb:greaterThanOrEqual(?s, umbral) \wedge swrlx:makeOWLThing(?x, ?w) -> probability(?x, "2.0"^^xsd:float) \wedge DeliberatedUnauthorizedAccess(?x) \wedge type(?x, "Threat Deliberated Unauthorized Access") \wedge impact(?x, "4.0"^^xsd:float) \wedge isGeneratedBy(?x, ?w) \wedge numType(?x, 14)}$

Tabla 23. Reglas de Anomalías: Anomalía Redes Móviles

### Reglas de Anomalías: Anomalía de Ciberseguridad

En este caso, la probabilidad de que exista una amenaza aumentará si varias anomalías de tipo Ciberseguridad coinciden en su valor de IP destino. Esto significa que las fuentes de datos tienen información de anomalías que dirigen tráfico a un dispositivo que tiene una IP determinada. Esta anomalía genera automáticamente una amenaza de tipo *Deliberated Unauthorized Access* mediante la siguiente regla.

REGLA	ANOMALIES CYBERSECURITY
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Deliberated Unauthorized Access</i> debido a la existencia de varias anomalías de tipo ciberseguridad, procedentes de un mismo dispositivo. Se crean amenazas si el valor del parámetro <i>suspicious value</i> supera un umbral.
FÓRMULA	Cybersecurity_Sensor_Anomaly(?w) ^ suspicious_value(?w, ?s) ^ swrlb:greaterThanOrEqual(?s, 4.0) ^ swrlx:makeOWLThing(?x, ?w) -> probability(?x, "2.0"^^xsd:float) ^ DenialOfService(?x) ^ type(?x, "Threat Denial Of Service") ^ impact(?x, "4.0"^^xsd:float) ^ isGeneratedBy(?x, ?w) ^ numType(?x, 15)

Tabla 24. Reglas de Anomalías: Anomalía Ciberseguridad

### Reglas de Inventario de Amenazas: Deliberated Information Leak Threat

REGLA	DELIBERATED INFORMATION LEAK THREAT
DESCRIPCIÓN	Esta regla permite generar automáticamente amenazas de tipo <i>Deliberated Information Leak</i> debido a la existencia de un dispositivo a través del cual se tiene acceso a datos clasificados y que está afectado por una vulnerabilidad.
FÓRMULA	Users(?u) ^ dependsOn(?cdata,?sw) ^ ClassifiedData(?cdata) ^ has_access_to(?u,?cdata) ^ dependsOn(?sw,?hw) ^ dependsOn(?hw,?u) ^ dependsOn(?hw,?lan) ^ dependsOn(?lan,?router) ^ Hardware(?router) ^ has_vulnerability(?router,?v) ^ Vulnerability(?v) ^ has_cyberobservable(?router,?mac) ^ MAC-Addr(?mac) ^ WiFi_Sensor_Anomaly(?w) ^ related-to(?w, ?mac) ^ swrlx:makeOWLThing(?x, ?v) -> DeliberatedInformationLeak(?x) ^ type(?x, "Deliberated Information Leak Threat") ^ impact(?x, "6.0"^^xsd:float) ^ probability(?x, "4.0"^^xsd:float) ^ numType(?x, 9) ^ threatens(?x, ?cdata) ^ exploits(?x, ?v)

Tabla 25. Reglas de Inventario de Amenazas: Deliberated Information Leak Threat

### Reglas de Inventario de Riesgos: Data Protection Risk

REGLA	DATA PROTECTION RISK
DESCRIPCIÓN	Esta regla permite generar automáticamente riesgos de tipo Data Protection Risk, asociados a amenazas de Data Protection.
FÓRMULA	DataProtectionRisks(?x) ^ probability(?x, ?p) ^ cyberthreat_DRM:dependsOn(?rs, ?a) ^ cibersituational-ontology:impact(?x, ?i) ^ swrlx:makeOWLThing(?r, ?x) ^ cyberthreat_DRM:threatens(?x, ?a) ^ cyberthreat_DRM:Risk_Scope(?rs) -> cyberthreat_DRM:threatens(?r, ?rs) ^ cyberthreat_DRM:DataProtectionComplianceRisk(?r) ^ cibersituational-ontology:type(?r, "Data Protection Compliance")

	Risk"^^rdf:PlainLiteral) ^ cyberthreat_STIXDRM:isGeneratedBy(?r, ?x) ^ cyberthreat_DRM:damages(?r, ?a)
--	---

Tabla 26. Reglas de Inventario de Riesgos: Data Protection Risk

### Reglas de Inventario de Riesgos: Deliberated Unauthorized Access Risk

REGLA	DELIBERATED UNAUTHORIZED ACCESS RISK
DESCRIPCIÓN	Esta regla permite generar automáticamente riesgos de tipo Deliberated Unauthorized Access. Este riesgo se produce debido a la existencia de una amenaza de tipo Deliberated Unauthorized Access.
FÓRMULA	$  \begin{aligned}  & \text{cyberthreat_ONA:WiFi_Sensor_Anomaly(?w)} \wedge \text{cibersituational-ontology:probability(?x, ?p)} \\  & \text{cyberthreat_DRM:DeliberatedUnauthorizedAccess(?x)} \\  & \text{cibersituational-ontology:type(?x, "Threat Deliberated Unauthorized Access")} \\  & \wedge \text{cibersituational-ontology:impact(?x, ?i)} \\  & \text{cyberthreat_DRM:Risk_Scope(?rs)} \wedge \text{swrlx:makeOWLThing(?r, ?x)} \rightarrow \\  & \text{cyberthreat_DRM:DeliberatedUnauthorizedAccessRisk(?r)} \\  & \wedge \text{cyberthreat_STIXDRM:isGeneratedBy(?r, ?x)} \\  & \wedge \text{cyberthreat_DRM:threatens(?r, ?rs)} \wedge \text{cibersituational-ontology:type(?r, "Deliberated Unauthorized Access Risk")}  \end{aligned}  $

Tabla 27. Reglas de Inventario de Riesgos: Deliberated Unauthorized Access Risk

### Reglas de Inventario de Riesgos: Denial of Service Risk

REGLA	DENIAL OF SERVICE RISK
DESCRIPCIÓN	Esta regla permite generar automáticamente riesgos de tipo Denial Of Service. Este riesgo se produce debido a la existencia de una amenaza de tipo Denial of Service.
FÓRMULA	$  \begin{aligned}  & \text{cibersituational-ontology:probability(?x, ?p)} \\  & \text{cyberthreat_DRM:DenialOfService(?x)} \wedge \text{cibersituational-ontology:type(?x, "Threat Denial Of Service")} \\  & \wedge \text{cibersituational-ontology:impact(?x, ?i)} \wedge \text{cyberthreat_DRM:Risk_Scope(?rs)} \\  & \text{swrlx:makeOWLThing(?r, ?x)} \rightarrow \\  & \text{cyberthreat_DRM:DenialOfServiceRisk(?r)} \\  & \wedge \text{cyberthreat_STIXDRM:isGeneratedBy(?r, ?x)} \\  & \wedge \text{cyberthreat_DRM:threatens(?r, ?rs)} \wedge \text{cibersituational-ontology:type(?r, "Denial of Service Risk")}  \end{aligned}  $

Tabla 28. Reglas de Inventario de Riesgos: Denial of Service Risk

### Reglas de Inventario de Riesgos: Deliberated Information Leak Risk

REGLA	DELIBERATED INFORMATION LEAK RISK
DESCRIPCIÓN	Esta regla permite generar automáticamente riesgos de tipo Deliberated Information Leak. Este riesgo se produce debido a la existencia de una amenaza de tipo Deliberated Information Leak.
FÓRMULA	$  \begin{aligned}  & \text{cyberthreat\_DRM:DeliberatedInformationLeak(?x)} \\  & \text{cibersituational-ontology:type(?x, "Deliberated Information Leak Threat")} \wedge \\  & \text{cibersituational-ontology:impact(?x, ?i)} \wedge \\  & \text{cibersituational-ontology:probability(?x, ?p)} \wedge \\  & \text{cyberthreat\_DRM:threatens(?x, ?a)} \wedge \\  & \text{cyberthreat\_DRM:Risk_Scope(?rs)} \wedge \\  & \text{cyberthreat\_DRM:dependsOn(?rs, ?a)} \wedge \\  & \text{swrlx:makeOWLThing(?r, ?x)} \rightarrow \\  & \text{cyberthreat\_DRM:DeliberatedInformationLeakRisk(?r)} \\  & \wedge \\  & \text{cibersituational-ontology:type(?r, "Deliberated Information Leak Risk")} \\  & \wedge \\  & \text{cibersituational-ontology:impact(?r, ?i)} \wedge \\  & \text{cibersituational-ontology:probability(?r, ?p)} \wedge \\  & \text{cyberthreat\_STIXDRM:isGeneratedBy(?r, ?x)} \wedge \\  & \text{cyberthreat\_DRM:threatens(?r, ?rs)}  \end{aligned}  $

Tabla 29. Reglas de Inventario de Riesgos: Deliberated Information Leak Risk

### Reglas de Evaluación de Riesgos: Residual Risk for Data Protection Compliance Risk

Estas reglas suponen la reducción del riesgo potencial automáticamente debido a la existencia de salvaguardas. El riesgo potencial se calcula de la siguiente manera:

$$\text{PotentialRisk} = \text{Impact} + \text{Probability} \quad (1)$$

La existencia de salvaguardas es conveniente para poder proteger el sistema. Según el modelo MAGERIT, una salvaguarda es un “procedimiento o mecanismo tecnológico que reduce el riesgo”. En primer lugar, es necesario establecer la relación entre la salvaguarda y la amenaza para que posteriormente, si existe dicha amenaza se puede utilizar la salvaguarda para contrarrestar los efectos negativos de la amenaza. En este caso se ha considerado que tener salvaguardas para proteger el Cumplimiento de la Protección de Datos es algo que cualquier organización debería tener y por eso se propone como ejemplo. En la ontología se enumera un gran número de salvaguardas. De ellas, se elige la salvaguarda de tipo Control de Privilegios de Datos. Hay que tener en cuenta que el coste de la salvaguarda debe de ser menor que el valor del activo que se defiende, sino no tendría sentido gastar recursos en algo que no tiene valor para la organización. Teniendo en cuenta la salvaguarda, la siguiente regla realiza el cálculo del riesgo residual, mediante la siguiente fórmula:

$$\text{ActualRisk} = \text{PotentialRisk} - \text{value} \quad (2)$$

El valor del parámetro *value* es el valor en que se reduce el riesgo gracias a la salvaguarda.

REGLA	RESIDUAL RISK FOR DATA PROTECTION COMPLIANCE RISK
DESCRIPCIÓN	Mediante esta regla el razonador deducirá el valor del Riesgo Residual para el Riesgo de tipo Cumplimiento de Protección de Datos

	(Data Protection Compliance). Para ello tendrá en cuenta las salvaguardas de tipo Control de Privilegios sobre los Datos (PrivilegeDataControl).
FÓRMULA	<pre> cyberthreat_DRM:threatens(?x, ?rs) ^ cibersituational- ontology:type(?s, "privacy-and-data-protection-control") ^ cibersituational-ontology:drm_value(?s, ?v) ^ cibersituational- ontology:potentialRisk(?x, ?prisk) ^ cibersituational-ontology:type(?x, &gt;Data Protection Compliance Risk"^^rdf:PlainLiteral) ^ swrlb:subtract(?ar, ?prisk, ?v) ^ cyberthreat_DRM:PotentialRisk(?x) ^ cyberthreat_DRM:Data(?new_data) ^ cyberthreat_DRM:dependsOn(?rs, ?new_data) ^ cyberthreat_DRM:PrivacyandDataProtectionControl(?s) -&gt; cyberthreat_DRM:isMitigatedBy(?x, ?s) ^ cyberthreat_DRM:ResidualRisk(?x) ^ cibersituational- ontology:actualRisk(?x, ?ar) </pre>

Tabla 30. Reglas de Evaluación de Riesgos: Residual Risk for Data Protection Compliance Risk

### Reglas de Evaluación de Riesgos: Residual Risk for Denial of Service Risk

Esta regla sigue un procedimiento análogo al anterior, pero para el caso de la existencia de un riesgo de tipo Denegación de Servicio.

REGLA	RESIDUAL RISK FOR DENIAL OF SERVICE RISK
DESCRIPCIÓN	Mediante esta regla el razonador deducirá el valor del Riesgo Residual para el Riesgo de tipo Denegación de Servicio (Denial of Service). Para ello tendrá en cuenta las salvaguardas de tipo SecurityControl.
FÓRMULA	<pre> cyberthreat_DRM:threatens(?x, ?rs) ^ cibersituational- ontology:type(?s, "spoofing-test-filters-control") ^ cibersituational-ontology:drm_value(?s, ?v) ^ cibersituational- ontology:potentialRisk(?x, ?prisk) ^ cibersituational-ontology:type(?x, "Denial of Service Risk") ^ swrlb:subtract(?ar, ?prisk, ?v) ^ cyberthreat_DRM:PotentialRisk(?x) ^ cyberthreat_DRM:Data(?new_data) ^ cyberthreat_DRM:dependsOn(?rs, ?new_data) ^ cyberthreat_DRM:SecurityFunctionalTestControl(?s) -&gt; cyberthreat_DRM:isMitigatedBy(?x, ?s) ^ cyberthreat_DRM:ResidualRisk(?x) ^ cibersituational- ontology:actualRisk(?x, ?ar) </pre>

Tabla 31. Reglas de Evaluación de Riesgos: Residual Risk for Denial of Service Risk

### Reglas de Evaluación de Riesgos: Residual Risk for Deliberated Unauthorized Access Risk.

REGLA	RESIDUAL RISK FOR DELIBERATED UNAUTHORIZED ACCESS RISK
DESCRIPCIÓN	Mediante esta regla el razonador deducirá el valor del Riesgo Residual para el Riesgo de tipo Acceso No Autorizado (Deliberated

	Unauthorized Access). Para ello tendrá en cuenta las salvaguardas de tipo Control de Acceso (Access Control).
FÓRMULA	$  \begin{aligned}  & \text{cyberthreat\_DRM:threatens(?x, ?rs)} \wedge \text{cibersituational-} \\  & \text{ontology:type(?s, "access-control") } \wedge \text{cibersituational-} \\  & \text{ontology:drm_value(?s, ?v)} \wedge \text{cibersituational-} \\  & \text{ontology:potentialRisk(?x, ?prisk)} \wedge \text{cibersituational-ontology:type(?x,} \\  & \text{"Deliberated Unauthorized Access Risk") } \wedge \text{swrlb:subtract(?ar, ?prisk,} \\  & \text{?v)} \wedge \text{cyberthreat\_DRM:PotentialRisk(?x)} \\  & \text{cyberthreat\_DRM:Data(?new_data)} \wedge \text{cyberthreat\_DRM:dependsOn(?rs,} \\  & \text{?new_data)} \wedge \text{cyberthreat\_DRM:AccessControl(?s)} \rightarrow \\  & \text{cyberthreat\_DRM:isMitigatedBy(?x, ?s)} \\  & \text{cyberthreat\_DRM:ResidualRisk(?x)} \wedge \text{cibersituational-} \\  & \text{ontology:actualRisk(?x, ?ar)}  \end{aligned}  $

Tabla 32. Reglas de Evaluación de Riesgos: Residual Risk for Deliberated Unauthorized Access Risk

#### Reglas de Evaluación de Riesgos: Residual Risk for Deliberated Information Leak Risk.

REGLA	RESIDUAL RISK FOR DELIBERATED INFORMATION LEAK RISK
DESCRIPCIÓN	Mediante esta regla el razonador deducirá el valor del Riesgo Residual para el Riesgo de tipo Deliberated Information Leak. Para ello tendrá en cuenta las salvaguardas de tipo Control de la Gestión de Claves (Key Management Control).
FÓRMULA	$  \begin{aligned}  & \text{cyberthreat\_DRM:threatens(?x, ?rs)} \wedge \text{cibersituational-} \\  & \text{ontology:type(?s, "key-management-control") } \wedge \text{cibersituational-} \\  & \text{ontology:drm_value(?s, ?v)} \wedge \text{cibersituational-} \\  & \text{ontology:potentialRisk(?x, ?prisk)} \wedge \text{cibersituational-ontology:type(?x,} \\  & \text{"Deliberated Information Leak Risk") } \wedge \text{swrlb:subtract(?ar, ?prisk, ?v)} \\  & \wedge \text{cyberthreat\_DRM:PotentialRisk(?x)} \\  & \text{cyberthreat\_DRM:Data(?new_data)} \wedge \text{cyberthreat\_DRM:dependsOn(?rs,} \\  & \text{?new_data)} \wedge \text{cyberthreat\_DRM:KeysManagementControl(?s)} \rightarrow \\  & \text{cyberthreat\_DRM:isMitigatedBy(?x, ?s)} \\  & \text{cyberthreat\_DRM:ResidualRisk(?x)} \wedge \text{cibersituational-} \\  & \text{ontology:actualRisk(?x, ?ar)}  \end{aligned}  $

Tabla 33. Reglas de Evaluación de Riesgos: Residual Risk for Deliberated Information Leak Risk

## 6 DESARROLLO DEL SISTEMA

---

En este apartado se detalla el proceso que se ha seguido para el desarrollo del sistema. En primer lugar, se ha implementado la ontología en el editor Protégé, definiendo sus clases, propiedades y relaciones. En segundo lugar, se han implementado los *parsers* para la integración de los datos de fuentes externas en la ontología. En tercer lugar, se ha llevado a cabo el cálculo del riesgo. Por último, se lleva a cabo la verificación de la consistencia de la ontología por parte del razonador y se infiere nuevo conocimiento. Posteriormente se visualizan los resultados mediante el desarrollo de una interfaz gráfica, la cual se detalla en el siguiente capítulo.

Cada una de las partes desarrolladas pueden consultarse en un repositorio de GitHub [34].

### 6.1 IMPLEMENTACIÓN EN PROTÉGÉ

Para la implementación de la ontología se ha utilizado la herramienta Protégé. Se ha elegido esta herramienta porque es un editor de ontologías que facilita la implementación de las mismas con una interfaz fácil de usar. Además, dispone de la integración con muchos *plugins*, en especial con el lenguaje SWRL y el razonador Pellet, lo que la hace muy diversa, completa y adecuada para el desarrollo de este trabajo.

Como se ha mencionado anteriormente, parte de este trabajo se basa en la tesis de Raúl Riesco [32]. La tesis propone el desarrollo de una ontología formada por dos: una ontología que recoge información sobre amenazas en formato STIX (CTI) y una ontología relacionado con el análisis y gestión de riesgos (DRM). Esta ontología se adapta en este trabajo incluyendo otra adicional: una ontología que recoge información sobre anomalías (ONA). Además, como el formato STIX es un formato vivo y en continua evolución, había sido actualizado durante la elaboración del trabajo, y, por ello, se ha actualizado la ontología CTI a la versión nueva 2.1.

La creación de estas ontologías se ha llevado a cabo en Protégé. Para instalarlo, hace falta disponer de un entorno en el que ya se haya instalado previamente Java. Para descargarlo se accede a la página oficial, <http://protege.stanford.edu>, y se descarga el paquete del sistema operativo correspondiente. En este caso, se ha elegido la última versión 5.5.0 de Protégé y el paquete correspondiente al sistema Mac OS.

Para comenzar a crear la ontología primero se establece el IRI. Un IRI es un Recurso Identificador Internacionalizado que identifica unívocamente a una ontología y extiende al Identificador Único Universal (URI) por lo que es compatible con él. Todos los elementos de la ontología se identifican con IRIs pero para abreviar su uso se usan prefijos. En la siguiente imagen se observa el IRI de la ontología final, en la parte superior, y, en la parte inferior, cada uno de los prefijos que aparecen en la ontología.

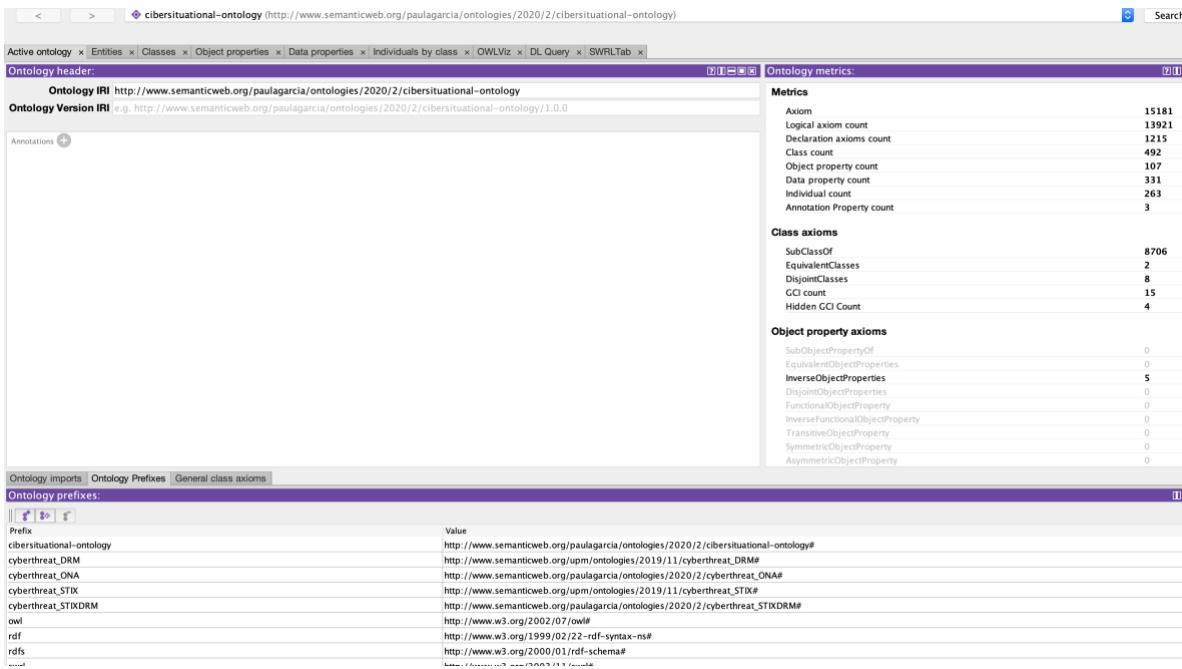


Figura 12. Definición de una ontología en el editor Protégé

Para entender cómo se definen las ontologías en Protégé se detallan algunos conceptos previos:

- **Classes:** son las entidades de un determinado dominio descritos formalmente. En la siguiente imagen se puede observar en amarillo las clases y subclases que forman la ontología ONA. En el recuadro de la derecha se especifican las características, propiedades y relaciones de cada una de las clases.

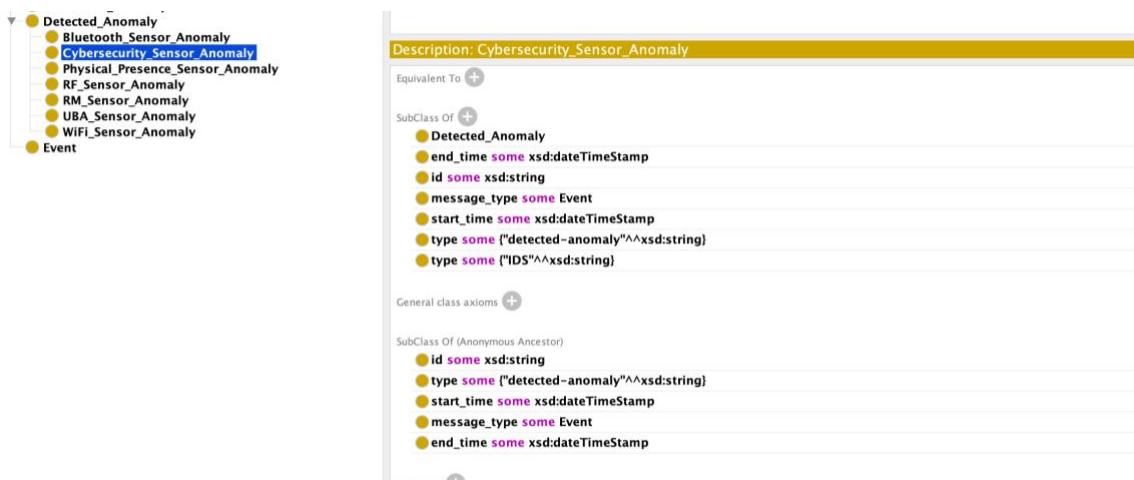


Figura 13. Clases de una ontología definidas en el editor Protégé

- **Datatype Properties:** son un tipo de propiedad que pueden tener las clases de una ontología y representan las características y atributos de cada una de ellas. Se caracterizan en que su rango será siempre un tipo de datos o *datatype*. En la siguiente imagen se observan algunas propiedades *datatype properties* que caracterizan cada una de las clases de la ontología ONA.

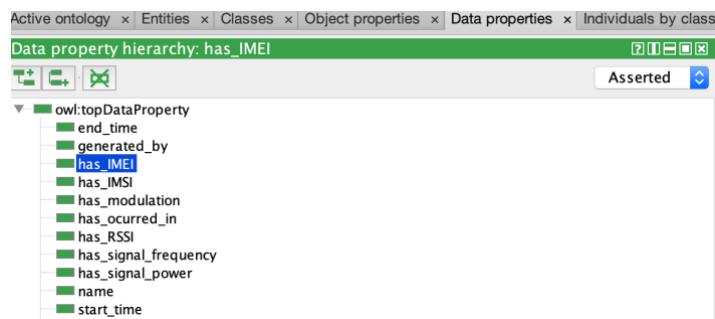


Figura 14. Datatype Properties de una ontología definidas en el editor Protégé

- **Object Properties:** son un tipo de propiedad que pueden tener las clases de una ontología y que representan las relaciones entre las distintas clases. Se caracterizan en que el tipo de su rango será una clase existente en la ontología. La siguiente imagen representa la creación de una propiedad *object property*. Además, en el cuadro de la derecha *Usage* se puede observar qué clases tienen esa propiedad.

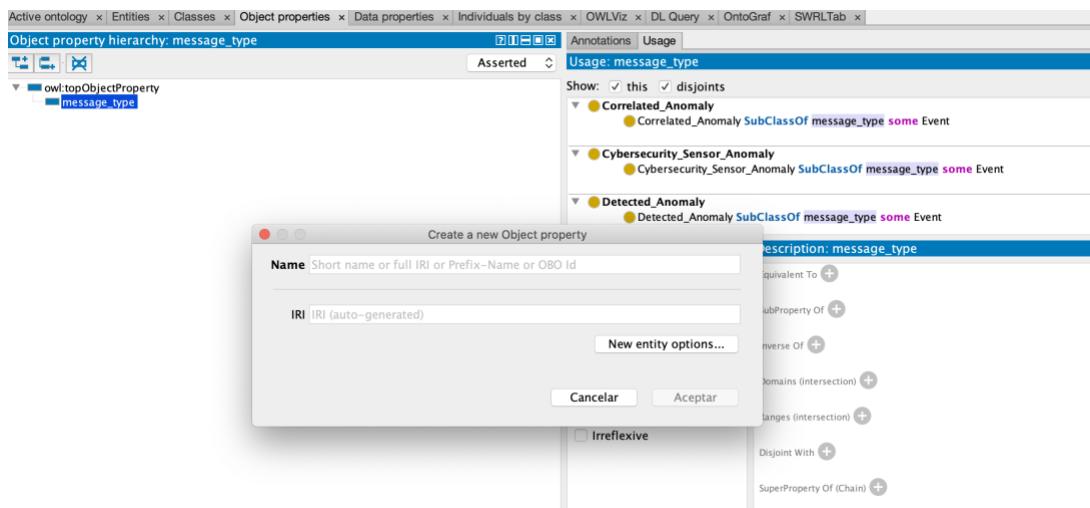


Figura 15. Object Properties de una ontología definidas en el editor Protégé

- **Datatypes:** constituyen los tipos de datos, *string*, *integer*, *float*, etc. La siguiente imagen muestra los tipos *datatypes* que vienen por defecto en la herramienta Protégé y a partir de los cuales se definen los atributos y características de cada una de las clases. Existen tipos owl procedentes del lenguaje OWL y de otros compatibles como rdf, rdfs o xsd, que corresponden con los lenguajes RDF, RDF-Schema y XML Schema.

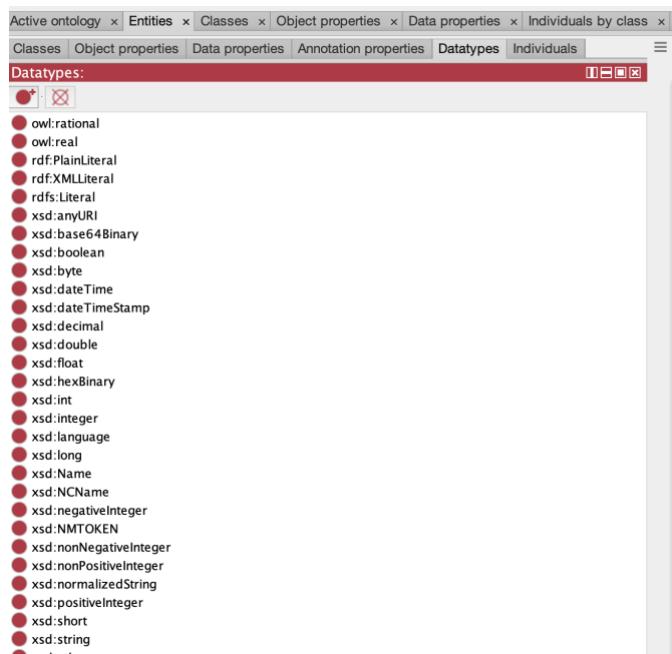


Figura 16. Datatypes de una ontología definidos en el editor Protégé

- **Individuals:** elementos identificables o instancias que se caracterizan por ser individuos concretos que representan la ontología y se asocian a cada una de las clases. En la siguiente imagen se observan instancias creadas en la ontología mediante la herramienta Protégé. En este caso, la instancia es un individuo que pertenece a la clase *Bluetooth\_Sensor\_Anomaly*. Como esta clase es subclase de *Detected\_Anomaly*, la instancia también se asocia con esta clase.

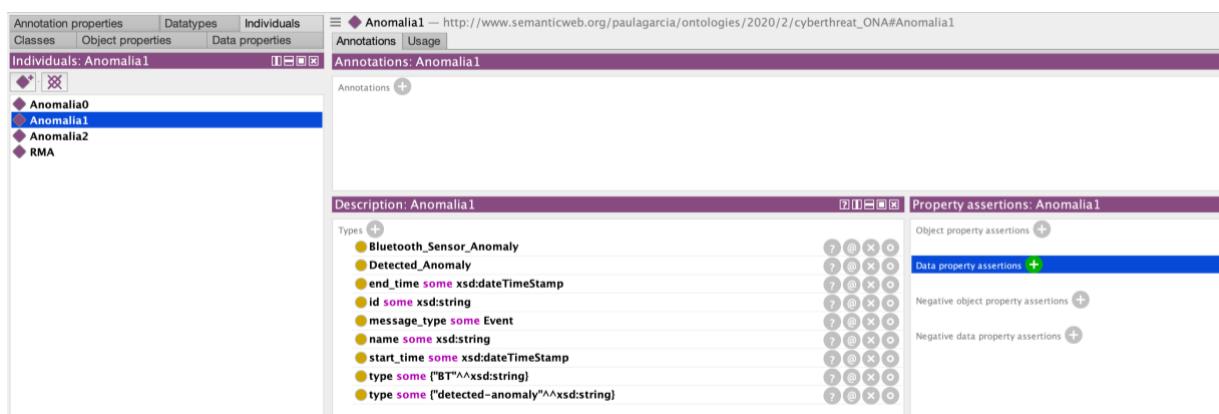


Figura 17. Individuals o instancias de una ontología definidos en el editor Protégé

El resultado de la definición de las clases, las propiedades y las relaciones de la ontología ONA se representa en la siguiente figura, mediante el plugin OntoGraph de Protégé.

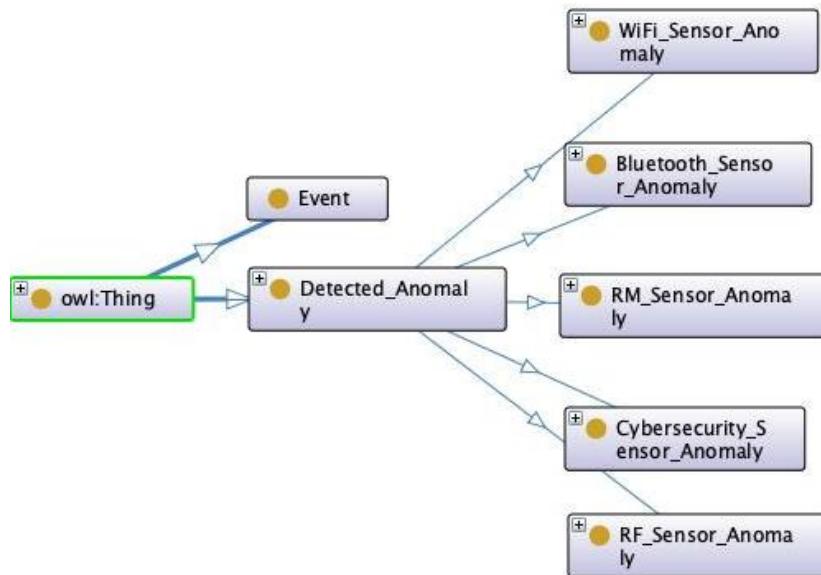


Figura 18. Representación de las clases de la ontología ONA mediante OntoGraph

Cabe destacar que cualquier ontología es siempre subclase de *owl:Thing* que es la clase principal.

Protégé permite la unión de ontologías independientes en una ontología final mediante lo que denomina *merge* de ontologías. Es por eso por lo que se han implementado las tres ontologías independientemente y, por último, se han unido para establecer las relaciones que existen entre ellas.

La herramienta Protégé incluye también herramientas para la visualización de los conceptos de las ontologías. De esta manera, es más sencillo visualizar las relaciones que se establecen entre cada uno de ellos, así como las instancias o ejemplares que pertenecen a cada clase. Estas herramientas se descargan mediante *plugins*, como *OWLviz* y *OntoGraph*.

En las siguientes figuras se representan las clases que constituyen cada una de las ontologías CTI y DRM mediante la herramienta *OntoGraph*.

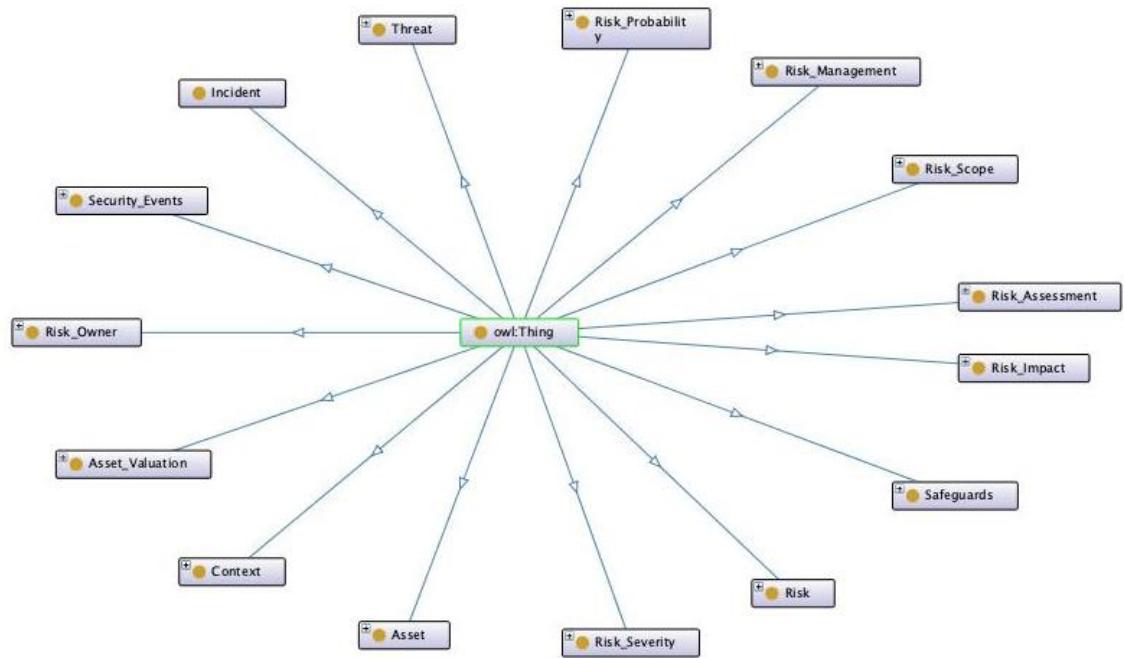


Figura 19. Clases de la ontología DRM

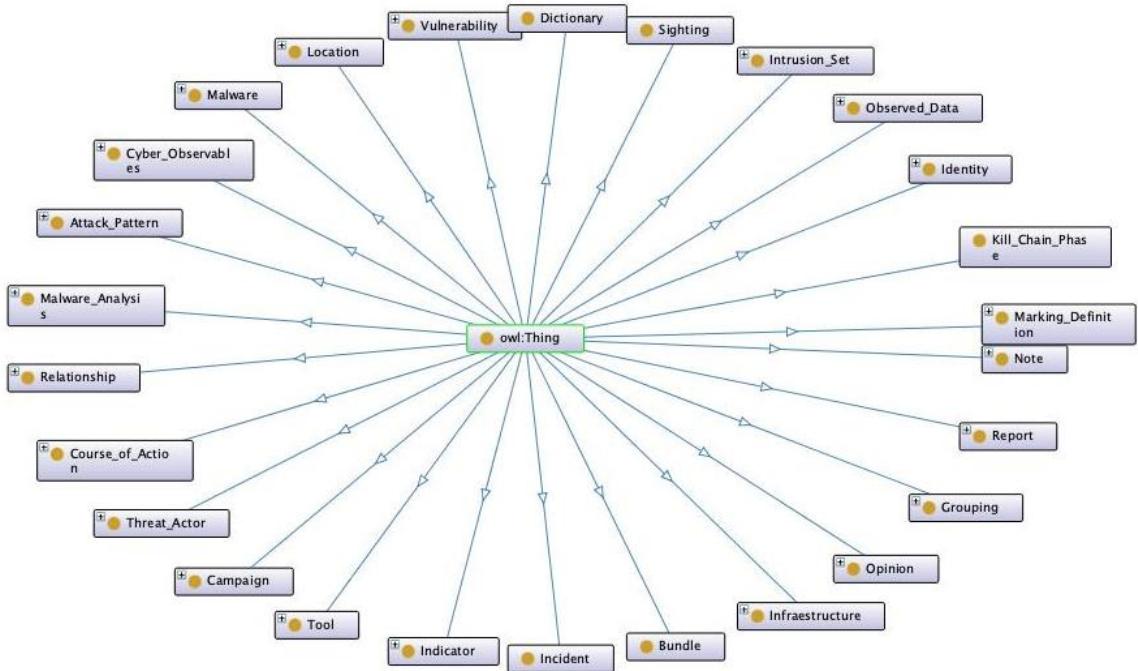


Figura 20. Clases de la ontología CTI

El resultado de la implementación en Protégé es una ontología final en el lenguaje OWL formada por casi 500 clases y más de 15.000 axiomas declarados.

## 6.2 IMPLEMENTACIÓN DE PARSERS PARA LA INTRODUCCIÓN DE DATOS EN LA ONTOLOGÍA

Protégé permite la creación de individuos o instancias y su asociación con las clases de la ontología. Sin embargo, esto debe realizarse de manera manual, añadiendo cada dato individualmente.

Uno de los requisitos en este trabajo es que el proceso que se lleva a cabo sea rápido para poder detectar cualquier posible amenaza lo antes posible. Por ello, cuanto más automatizado sea el proceso, más rápido se podrá integrar la nueva información en la ontología. Esta integración será eficiente y consistente una vez comprobada por el razonador. Como se ha elegido el razonador Pellet y según lo comentado en el apartado 3.3.4.2, el proceso de inferencia es muy rápido y con un tiempo de respuesta bajo.

Para la implementación de los *parsers* que se encargan de introducir los datos en la ontología según una estructura definida, se necesitan las siguientes herramientas. Cabe destacar que, puesto que OWLAPI, SWRL y Pellet tienen un proceso de evolución independiente, se han encontrado conflictos a la hora de utilizar las últimas versiones para poder dar una solución lo más actualizada posible. Sin embargo, se ha tenido que reajustar cada una de las versiones según la compatibilidad entre las herramientas. Se han utilizado:

- Java 8.
- Eclipse.
- Para leer y modificar la ontología se ha utilizado la OWL API, en concreto la versión OWL API 4.5.9.
- Razonador Pellet 2.2.0.
- SWRL API, versión 2.0.8.

En primer lugar, es necesario cargar la ontología. Para ello, antes se debe crear una instancia de *OWLontologyManager*, que se encarga de gestionar la ontología. Se utiliza para crear, cargar y acceder a las ontologías. Una vez se ha cargado la ontología, se obtiene su IRI para poder acceder a cada una de las clases y propiedades creadas.

```
OWLontologyManager man = OWLManager.createOWLontologyManager();
OWLontology o = man.loadOntologyFromOntologyDocument(file);
OWLDataFactory dataFactory = man.getOWLDataFactory();
IRI documentIRI = o.getOntologyID().getOntologyIRI().get();
```

Un ejemplo de IRI es el siguiente, que corresponde a la clase *Detected Anomaly*:

[http://www.semanticweb.org/paulagarcia/ontologies/2020/2/cyberthreats\\_ONA#Detected\\_Anomaly](http://www.semanticweb.org/paulagarcia/ontologies/2020/2/cyberthreats_ONA#Detected_Anomaly)

La información procedente de las fuentes externas pasa por los *parsers* donde se extraen los datos que posteriormente son incluidos en la ontología como instancias o individuos. A continuación, se detallan cada uno de los *parsers* implementados.

---

### 6.2.1 PARSER PARA LAS FUENTES DE DATOS DE ANOMALÍAS

---

La información de anomalías de las fuentes externas se almacena en el sistema de almacenamiento. Estos datos proceden de fuentes fiables que determinan si existe una anomalía y almacenan la información en el sistema.

Para añadir dicha información se ha creado una clase de Java llamada *Anomaly* mediante la cual los datos se incorporan a la ontología, identificando la clase a la que pertenecen, sus propiedades y relaciones con otras clases. Las clases a las que se asocian los datos son las definidas en el apartado Clases de la ontología y son las pertenecientes a la ontología ONA.

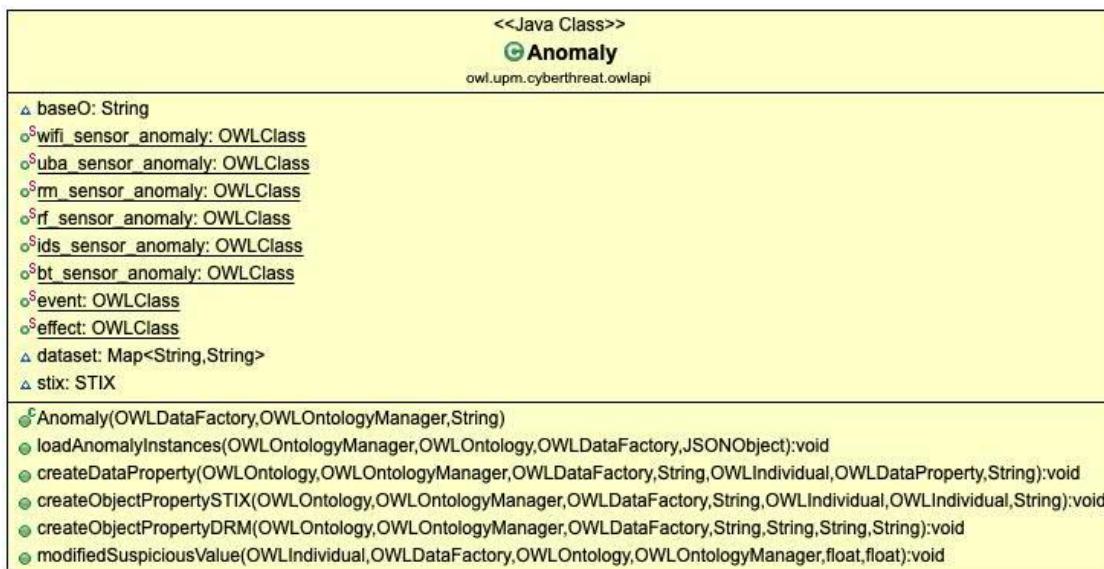


Figura 21. Diagrama UML para la clase Anomalies.java

Añadir la información de anomalías significa crear instancias o individuos y sus correspondientes IRIs. Para ello, se hace uso de la clase *OWLIndividual* de OWL API, a partir de la cual se crea una instancia. Cuando se crea una instancia o individuo se asocia con la clase a la que pertenece mediante declaración de axiomas.

Como se ha dicho anteriormente, un axioma es una sentencia que especifica las relaciones que deben cumplir los elementos de la ontología. Mediante los axiomas se relacionan individuos, se definen sus propiedades y sus relaciones.

```

OWLIndividual anomaly_instance =
dataFactory.getOWLNamedIndividual(IRI.create(documentIRI.toString()
+"#" +name));
OWLClassAssertionAxiom axioma =
dataFactory.getOWLClassAssertionAxiom(rm_sensor_anomaly,
anomaly_instance);
man.addAxiom(o, axioma);

```

Con el ejemplo anterior se muestra la creación de una instancia o ejemplar de anomalía de tipo Redes Móviles que se asocia a la clase correspondiente.

Para manejar los datos dentro de la ontología una vez cargada, se utiliza *OWLDataFactory* que permite crear entidades, expresiones de clase y axiomas. A cada una de las instancias se asocian sus correspondientes propiedades y atributos. Para ello, se hace uso de *OWLObjectProperty* y *OWLDataProperty*. Los métodos *createDataProperty(...)*, *createObjectPropertySTIX(...)* y *createObjectPropertyDRM(...)* permiten generar automáticamente los axiomas correspondientes para crear dichas propiedades.

Cuando se incorporan las anomalías a la ontología, se establece un valor por defecto del parámetro *suspicious\_value*. Como ya se ha mencionado, las anomalías pueden generar por sí mismas amenazas o, en caso de existir ya amenazas en el sistema, si se dan las características adecuadas, pueden incrementar un riesgo. Para ello, se utiliza el parámetro *suspicious\_value*. Este parámetro se incrementa si existen anomalías similares, cuya existencia puede aumentar la posibilidad de que ocurra una amenaza.

El parámetro *suspicious value*, se incrementa según un valor que se determina mediante un parámetro llamado *intervalo*. Esto se realiza mediante el método *modifiedSuspiciousValue()*. El parámetro *intervalo* se configura antes de iniciarse la ejecución del programa. Lo mismo ocurre con el parámetro llamado *umbral*, el cual establece el valor mínimo del parámetro *suspicious\_value* a partir del cual una anomalía genera una amenaza. De la misma manera, el parámetro *umbral* se configura previamente a la ejecución del programa.

---

#### 6.2.2 PARSER PARA LAS FUENTES DE DATOS DE THREAT INTELLIGENCE

---

La información de *Threat Intelligence* procedente de fuentes externas se almacenan de la misma manera que en el caso de anomalías. En este caso, la información sigue la estructura STIX mediante un formato JSON.

Para integrar los datos en la ontología se ha creado una clase de Java llamada *STIX*. Esta clase es la que se encarga de mapear los datos a cada una de las clases que pertenecen a la ontología CTI y determina cada una de las propiedades y relaciones en función de dichos datos.



Figura 22. Diagrama UML para la clase STIX.java

Esta clase también incluye métodos que generan automáticamente la declaración de los axiomas para cada una de las instancias o individuos de STIX.

Gracias a que los datos de *Threat Intelligence* siguen la estructura STIX se puede compartir información con otras organizaciones que sigan la misma estructura estándar. Esta información es información de inteligencia de amenazas y es fundamental para poder tener una respuesta proactiva ante incidentes y ataques y enriquecer la conciencia cibersituacional.

### 6.2.3 PARSER PARA LOS ACTIVOS PROCEDENTES DE LA HERRAMIENTA PILAR

Cualquier organización necesita bienes y recursos para poder llevar a cabo su actividad económica y obtener unos beneficios. Estos bienes y recursos se identifican como activos de la empresa y es necesario tenerlos en cuenta a la hora de realizar la gestión de riesgos.

La generación de la información de los activos se ha realizado mediante la herramienta PILAR. Se realiza una valoración por dependencias. Los activos y sus dependencias se exportan de PILAR en formato CSV y la valoración de los mismos se exporta en formato XML. El valor acumulado debido a las dependencias entre los activos no se necesita puesto que se realiza mediante la ejecución de las reglas mencionadas en 5.2.1, llamadas ‘Reglas de Valoración de Activos’.

Para integrar los datos en la ontología se ha creado una clase de Java llamada *DRM*. Esta clase es la que se encarga de mapear los datos de activos a cada una de las clases a las que pertenecen de la ontología DRM.



Figura 23. Diagrama UML para la clase DRM.java

Por un lado, se crean las instancias correspondientes a los activos mediante los métodos `createAssets(...)`, y con los métodos `createDependencies(...)` y `createAssetValuation(...)` se constituyen las dependencias que se establecieron en PILAR y se crea la valoración de los activos que se asocia a un *Risk Scope*, como se explicará en el siguiente capítulo.

El resto de los métodos de la clase DRM son los necesarios para gestionar las amenazas y riesgos.

Una vez que la información de fuentes externas es modelada en OWL, las reglas SWRL definidas en 5.2 se ejecutan y el razonador verifica la consistencia de la ontología e infiere nuevo conocimiento sobre la base de conocimiento de la ontología.

### 6.3 CÁLCULO DINÁMICO DEL RIESGO

Otro de los requisitos de este trabajo es que se debe calcular el riesgo de manera dinámica. Para lo cual se tienen en cuenta las medidas realizadas en el pasado y se observa la evolución del riesgo. Para conseguirlo, se ha estudiado el caso que se describe en [35] y se ha adaptado al presente trabajo. Para su desarrollo se han utilizado las herramientas del apartado anterior (Java8, Eclipse, etc.).

En [35] se modela el riesgo de distintos dominios administrativos en tiempo real. Se establece España como objeto de cálculo del riesgo. La idea es que se calcula el riesgo por dominios. Primero por Comunidades Autónomas y después en el conjunto de toda España. Además, el sistema dispone de una capacidad de memoria para poder realizar un cálculo continuo del riesgo teniendo en cuenta los cálculos del riesgo anteriores.

En el caso de este trabajo, el cálculo del riesgo se realiza a nivel de la organización. Las reglas SWRL realizan el proceso de análisis y evaluación de riesgos. En este proceso se calculan

automáticamente los riesgos potencial y residual que se generan como consecuencia de cada una de las amenazas y salvaguardas existentes en la organización.

Los valores del riesgo procedentes de las reglas son valores individuales de cada uno de los riesgos debido a la materialización de amenazas específicas. Es decir, debido a la existencia, por ejemplo, de una amenaza de Denegación de Servicio, se produce un riesgo de Denegación de Servicio y se evalúa generando los valores potenciales y residuales de ese riesgo.

Por tanto, cada riesgo  $k$  tiene un riesgo potencial y un riesgo residual, que puede ser cero o no dependiendo de si existen salvaguardas específicas para dicho riesgo  $k$ . La suma de los valores de cada uno de los riesgos  $k$  constituye el riesgo potencial total y el riesgo residual total de la organización en ese instante de tiempo  $t_i$ . Además, también se tiene en cuenta los pesos del número de las amenazas existentes de cada riesgo  $k$  puesto que cuanto mayor sea ese número mayor será su representación en el cálculo del riesgo total. Basta con aplicar las siguientes ecuaciones, donde  $t_i$  es el instante de tiempo actual:

$$\text{Potential Total Risk}(t_i) = \frac{\sum_k \text{namenazas}_k * \text{potentialRisk}_k}{\sum_k \text{namenazas}_k} \quad (3)$$

$$\text{Residual Total Risk}(t_i) = \frac{\sum_k \text{namenazas}_k * t_i * \text{residualRisk}_k}{\sum_k \text{namenazas}_k * t_i} \quad (4)$$

Estos resultados son discretos y pertenecen a un instante de tiempo determinado. Sin embargo, el cálculo del riesgo total no es independiente del cálculo realizado en otros instantes. Por lo que para poder dar un resultado lo más preciso y exacto posible, se realiza un cálculo en el que se tiene en cuenta los cálculos realizados en instantes anteriores al instante actual. El resultado ya no es discreto.

Ahora bien, si el tiempo aumenta, también lo hará el número de cálculos realizados. Por lo que para evitar que el crecimiento de los cálculos se vuelva insostenible se hace uso de una función, de la misma manera que en [35], con el fin de seleccionar una ventana de valores. Por tanto, solo se tendrán en cuenta los valores dentro de esa ventana de valores para calcular el riesgo potencial total final y el riesgo residual total final.

La caracterización de la función sigue el mismo procedimiento explicado en [35] donde se establece un parámetro de penalización, llamado tiempo de olvido, que determina el valor a partir del cual se descartan los resultados a tener en cuenta en el cálculo de los riesgos finales. Cabe destacar que el valor del parámetro de *penalización* lo establece el administrador o usuario del sistema en un fichero de configuración. Este valor se establece en minutos.

Para el cálculo de los riesgos totales finales basta con aplicar las siguientes fórmulas:

$$\text{Final Total Potential Risk}(t_i) = \frac{\sum_k nTotalamenazas_k * f(t_k) * \text{PotentialTotalRisk}_k}{\sum_k nTotalamenazas_k * t_k} \quad (5)$$

$$\text{Final Total Residual Risk}(t_i) = \frac{\sum_k nTotalamenazas_k * f(t_k) * \text{ResidualTotalRisk}_k}{\sum_k nTotalamenazas_k * t_k} \quad (6)$$

donde  $f(t)$  es la función de penalización, cuyo valor en el instante actual  $t_0$  es 1. La función  $f(t)$  es la utilizada en [35] y es una función exponencial caracterizada por la siguiente ecuación:

$$e^{\frac{-4t^3}{\delta^3}} \quad (7)$$

Los resultados se almacenan en un fichero JSON siguiendo un formato determinado. Estos resultados son los datos de entrada para la interfaz de visualización del sistema.

## 6.4 RAZONAMIENTO SEMÁNTICO

El razonador es el componente principal del sistema y es el responsable de inferir nuevo conocimiento sobre la ontología propuesta. Teniendo en cuenta las políticas y reglas definidas y los ejemplares o instancias que representan la información de contexto, los activos, las anomalías, etc., el razonador ejecuta el proceso por el cual analiza y evalúa los riesgos para su posterior gestión y cálculo.

Después del análisis realizado en el apartado de *Razonadores semánticos*, se elige Pellet como razonador semántico. Pellet es compatible con los lenguajes de definición de ontologías y reglas OWL, OWL2 y SWRL y tiene un tiempo de respuesta bajo. Esto es fundamental puesto que cuanto más rápido sea el procesamiento de la ontología, más rápido será el cálculo del riesgo que será más fiable porque estará más cerca del tiempo real.

El razonador supone una serie de beneficios en el sistema. Un razonador semántico se puede considerar la inteligencia del sistema, puesto que es él el que analiza y procesa los datos de la ontología semánticamente:

- Descubre cuándo algo no es consistente para poder corregir los errores y genera una serie de trazas para encontrar el origen del problema (*debugging*).
- Realiza una clasificación automática de las clases y subclases de la ontología, así como de las instancias o ejemplares nuevos que se introducen en ella. Para ello, hace uso de algoritmos de la lógica descriptiva.
- Ofrece una alta expresividad, para describir detalladamente conceptos complejos como las TTP de los atacantes, que como se ha dicho antes son fundamentales para saber su comportamiento y poder responder de manera más proactiva a los ataques.

El razonador trabaja sobre una base de conocimiento, la cual va enriqueciendo con las características anteriores.

En cuanto al significado de la ontología, uno de los objetivos es la capacidad de poder intercambiar información de amenazas con el resto de las organizaciones. La información que surge de la ontología, una vez ha sido procesada por el razonador, es una información fiable, que no tiene inconsistencias y que además sirve de apoyo a la conciencia cibersituacional.

## 7 VALIDACIÓN

---

Con el fin de comprobar las capacidades del sistema implementado, se somete a una serie de pruebas de validación.

En primer lugar, antes de la puesta en marcha del sistema, se deben completar unos ficheros de configuración. Estos ficheros incluyen todo lo que tiene que ver con las rutas al fichero de ontologías y a los sistemas de almacenamiento, así como los valores de los parámetros *UMBRAL*, *INTERVALO* y *PENALIZACIÓN* para que sea el usuario del sistema o el administrador el que indique el rango de valores según la precisión que requiera.

A continuación, se procede a validar el sistema. Para ello, se lleva a cabo un ejemplo de aplicación, donde se han identificado una serie de activos en un *Risk Scope* y se integran al mismo una serie de datos procedentes de fuentes externas. Se calcula el riesgo que supone la existencia de estos datos en el *Risk Scope* y se visualizan mediante una interfaz.

### 7.1 INTEGRACIÓN DE INFORMACIÓN DE ACTIVOS.

En primer lugar, se define el alcance o marco de estudio del plan de análisis y gestión de riesgos. Esto es a lo que se llama *Risk Scope* definido en [32] en la ontología DRM. El *Risk Scope* incluye todos los servicios y procesos que se encuentran dentro del alcance de un plan de riesgos determinado de la organización.

Para poder llevar a cabo los servicios y proyectos de la organización se necesita una serie de recursos o activos. Por ello, se identifican cada uno de los activos que forman parte del *Risk Scope* objeto de estudio. Una valoración de activos evalúa un *Risk Scope*, y por consiguiente todos los activos de los que depende.

En este caso se identifican un activo esencial y unos activos de soporte que dependen del mismo. Una vez identificados los activos se lleva a cabo la valoración, en concreto, una valoración por dependencias de manera que los activos relacionados con SW, HW, personal, etc., heredan el valor de los activos esenciales (en este caso Datos Clasificados). Con esta información se generan en PILAR los ficheros que contienen el inventario de activos y su valoración. Solo por el mero hecho de existir dichos activos, existen amenazas y como consecuencia, riesgos.

Para identificar estos activos se hace uso de la herramienta PILAR, que permite identificar y valorar activos siguiendo las normas y la metodología MAGERIT. De esta identificación se obtienen los activos y la valoración únicamente de los activos esenciales. Esta información se incluye a través del *parser* implementado en 6.2.3.

Para que el resto de los activos herede el valor de los activos esenciales se hace uso de las reglas llamadas ‘Reglas de Valoración de Activos’. Esto se hace así porque lo que se quiere es que el proceso sea lo más dinámico posible y si se incluyen nuevos activos no sea necesario realizar la valoración de nuevo de todos.

En las siguientes imágenes se puede observar la creación del *Risk Scope* y la correcta integración del activo *ClassifiedData*, con código ‘0’ y su relación de dependencia con el activo Software (dependsOn).

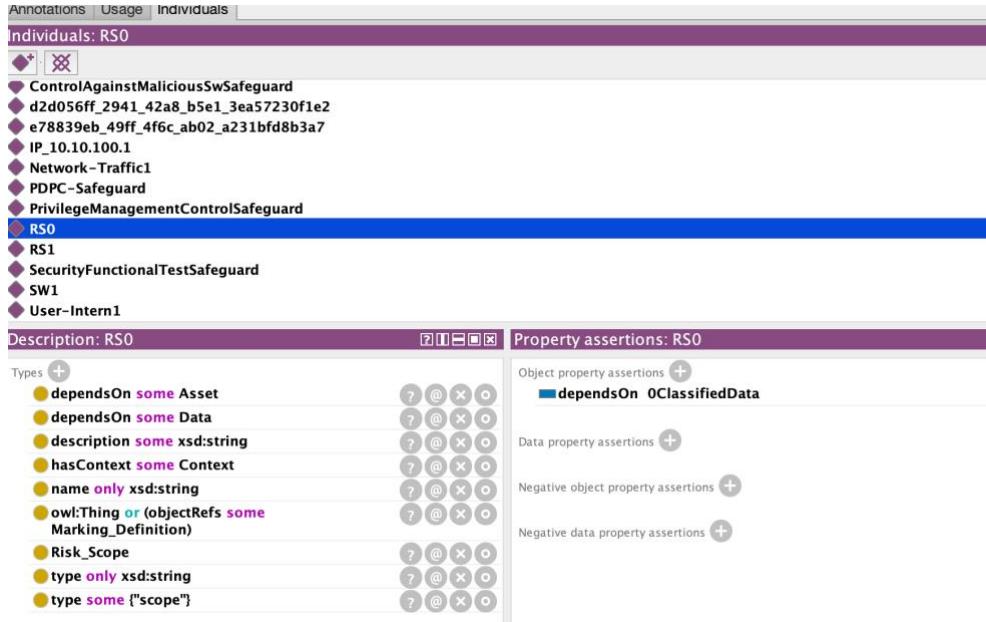


Figura 24. Creación de una Instancia de Risk Scope

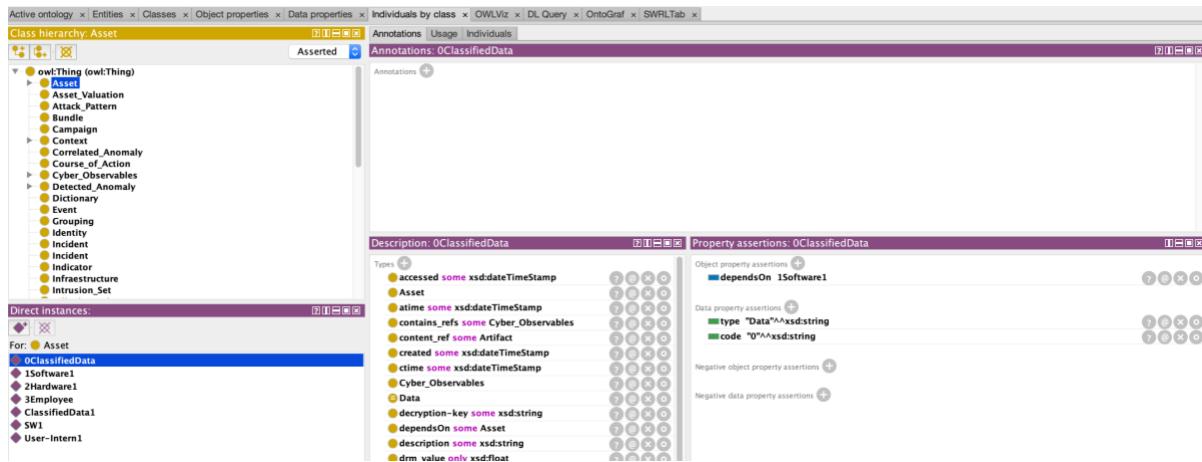


Figura 25. Prueba de validación correspondiente a la Integración de información de activos de PILAR

El *Risk Scope* depende de una serie de activos y cada uno de los cuales es evaluado mediante una valoración de activo. Para ello, se crean instancias de valoraciones de activos que se asocian con sus respectivos activos.

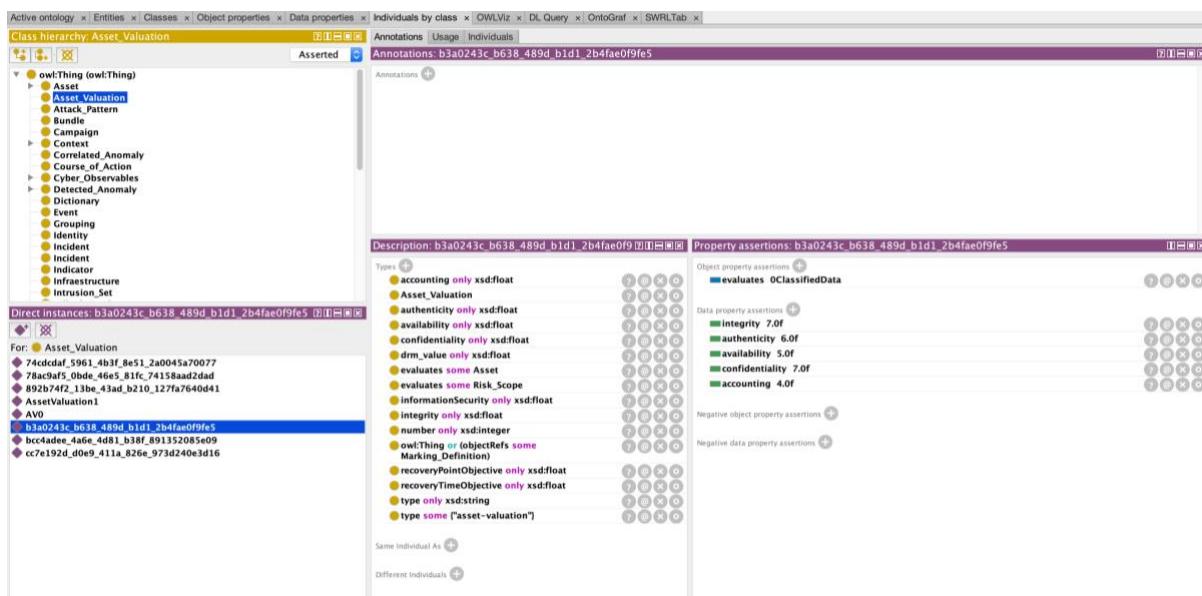


Figura 26. Prueba de validación correspondiente a la Integración de información de valoración de activos

A partir de esta valoración y las reglas mencionadas, se generan cada una de las valoraciones del resto de activos que dependen del activo esencial. La valoración del activo *Hardware* es la que se observa en la Figura 27.

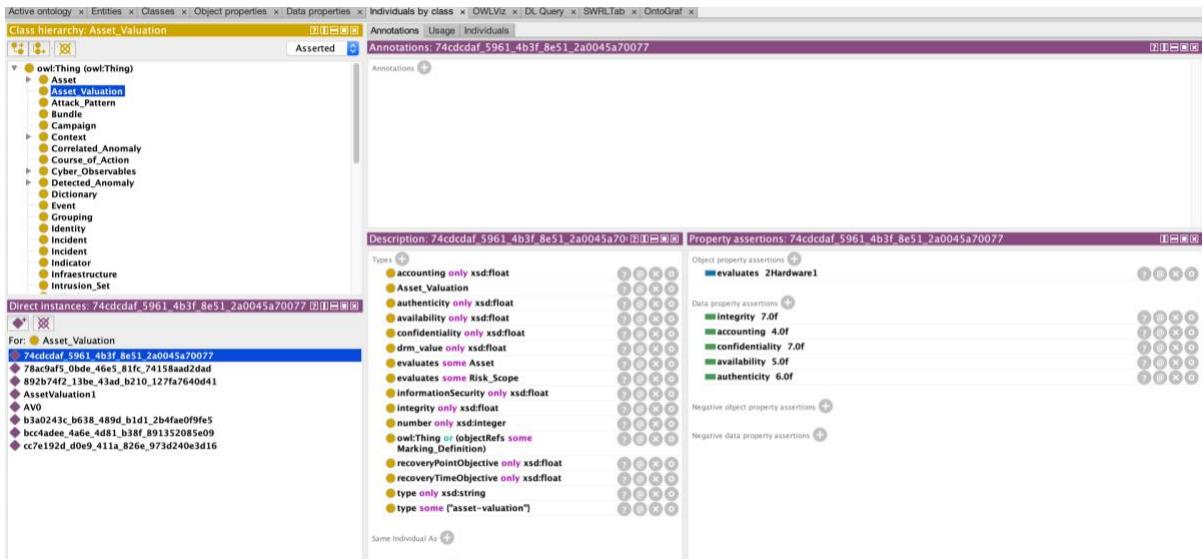


Figura 27. Prueba de validación correspondiente a la generación de valoraciones de activos derivadas del activo esencial

Al final, la asociación entre activos y valoraciones queda como se puede observar en la siguiente figura, donde la línea de color amarillo representa la dependencia entre los activos (relación *dependsOn*).

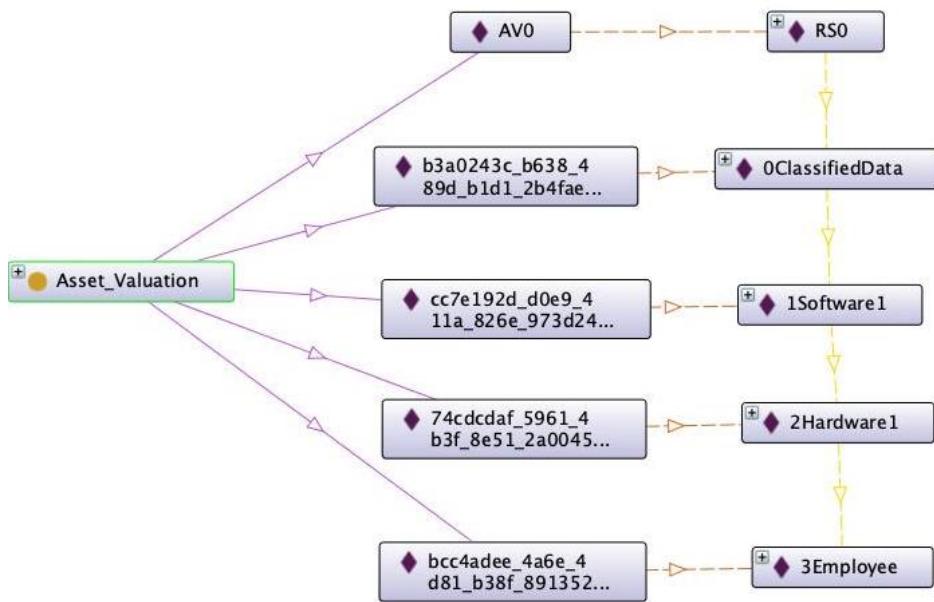


Figura 28. Asociación entre activos y valoración de activos

## 7.2 INTEGRACIÓN DE INFORMACIÓN DE ANOMALÍAS PROCEDENTE DE FUENTES EXTERNAS.

Cuando el sistema detecta la existencia de información en el sistema de almacenamiento, procede a introducirla en la ontología automáticamente. La información pasa a través del *parser* implementado en 6.2.1 y se mapea al lenguaje OWL para que sea procesada posteriormente por el razonador. La información de anomalías tiene la siguiente estructura:

```
[{
  "version": "1.0",
  "time": "20200126092709",
  "id": "001",
  "type": "WF",
  "event": "DATA",
  "anomaly": 1,
  "data": [
    {
      "userid": "00:00:00:00:00:01",
      "pwr": "400",
      "essid": "Cisco"
    }
  ]
},
{
  "version": "2.0",
  "time": "20200126092709",
  "id": "002",
  "type": "WF",
  "event": "DATA",
  "anomaly": 2,
  "data": [
    {
      "userid": "00:00:00:00:00:02",
      "pwr": "300",
      "essid": "Linksys"
    }
  ]
}]
```

```

    "type": "BT",
    "event": "DATA",
    "anomaly": 1,
    "data": [
      {
        "address": "00:00:ef:00:56:99:01",
        "rss": "29.0"
      }
    ]
},
{
  "version": "2.0",
  "time": "20200126092709",
  "id": "003",
  "type": "RM",
  "event": "DATA",
  "anomaly": 1,
  "data": [
    {
      "rat": "3G",
      "imei": "354096207395186",
      "imsi": "310150123456776"
    }
  ]
},
{
  "version": "2.0",
  "time": "20200126092709",
  "id": "004",
  "type": "RF",
  "event": "DATA",
  "anomaly": 1,
  "data": [
    {
      "signal": "-42",
      "freq": "433.920",
      "mod": "OOK"
    },
    {
      "signal": "-58",
      "freq": "433",
      "mod": "OOK"
    }
  ]
},
{
  "version": "2.0",
  "time": "20200126092709",
  "id": "005",
  "type": "IDS",
  "event": "DATA",
  "anomaly": 1,
  "data": [
    {
      "prediction": "No",
      "srcip": "59.166.0.7",
      "sport" : "46430",
      "dstip": "192.168.1.100",
      "dport": "443"
    }
  ]
}

```

```

        "dstip" : "149.171.126.1",
        "dsport" : "47200",
        "proto" : "tcp",
        "_id" : "AXjqjw0193k",
    }
]
}
]

]

```

En la Figura 29 se puede observar cómo se crea una nueva instancia de la clase *Detected Anomaly*, a la que se asocian las propiedades correspondientes en función de los datos de las fuentes externas. En este caso, la instancia corresponde a una anomalía de tipo Wifi. Esta instancia presenta la información a través de sus *Data Properties*, como *start\_time*, *pwr*, *has\_essid* o *suspicious\_value*. Para mostrar su relación con otras clases, se definen *Object Properties*, por ejemplo, para establecer la relación de su dirección MAC con la clase MAC-Addr de la ontología CTI.

La Figura 30 muestra la creación de una instancia de anomalía de tipo Ciberseguridad, incluyendo sus propiedades y características de la misma manera que en el caso anterior.

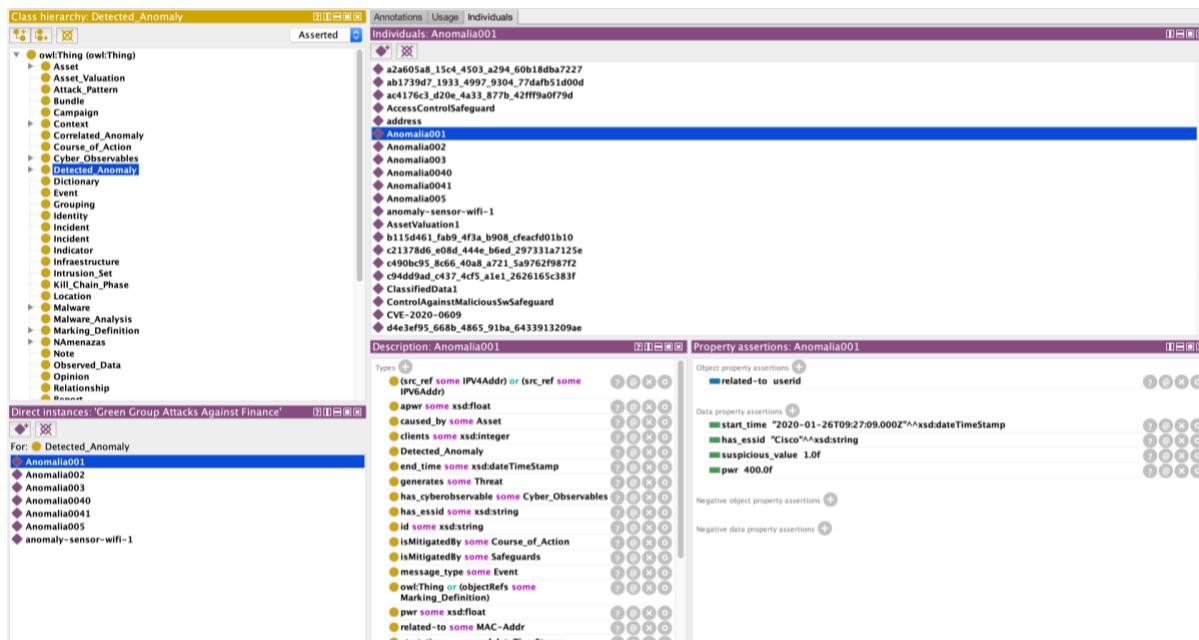


Figura 29. Prueba de validación correspondiente a la Integración de información sobre una anomalía de tipo Wifi en el sistema.

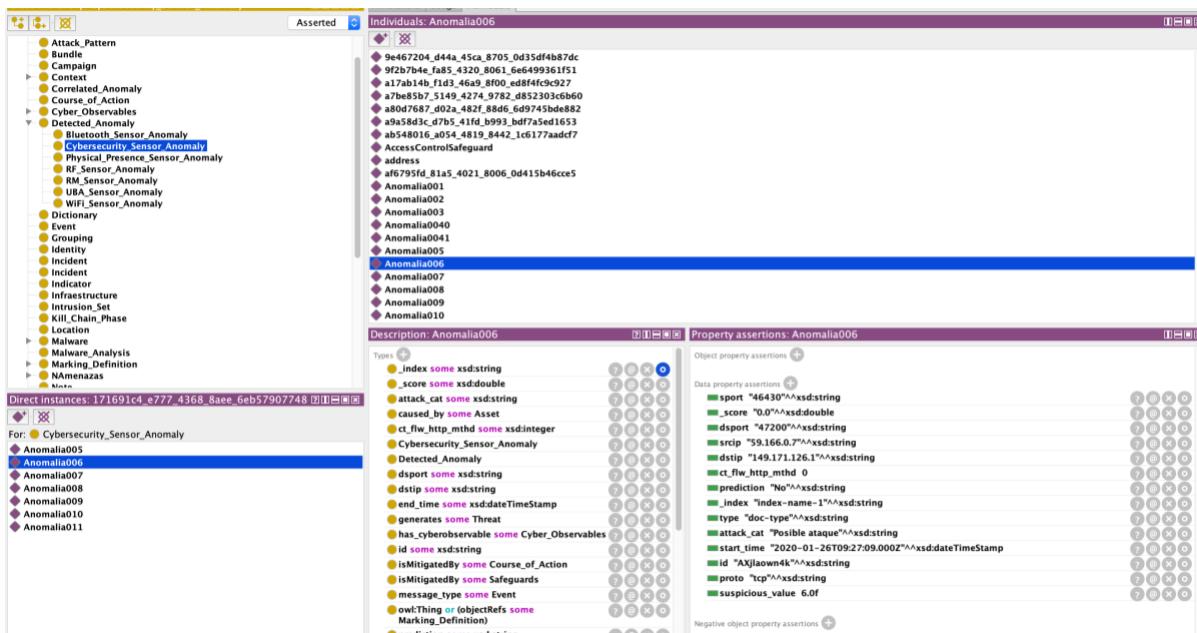


Figura 30. Prueba de validación correspondiente a la Integración de información sobre una anomalía de tipo Ciberseguridad en el sistema.

Como se observa se verifica la correcta integración de información de anomalías en la ontología.

### 7.3 INTEGRACIÓN DE INFORMACIÓN PROCEDENTE DE FUENTES DE THREAT INTELLIGENCE.

Como se ha mencionado antes, el formato de esta información es STIX. Una vez que el sistema detecta la existencia de nueva información en el sistema de almacenamiento, se introduce en el sistema. Se comprueba que se integra correctamente información de distintos elementos STIX, por ejemplo, datos sobre una Campaña o una Vulnerabilidad. Estos son algunos ejemplos de los objetos pertenecientes al dominio STIX. La información de TI tiene la siguiente estructura:

```
[{
    "type": "campaign",
    "spec_version": "2.1",
    "id": "campaign--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:03:00.000Z",
    "modified": "2016-04-06T20:03:00.000Z",
    "name": "Green Group Attacks Against Finance",
    "description": "Campaign by Green Group against a series of targets in the financial services sector."
},
{
    "type": "vulnerability",
    "spec_version": "2.1",
    "id": "vulnerability-0c7b5b88-8fff7-4a4d-aa9d-feb398cd0061",
    "created": "2020-05-02T08:17:27.000Z",
    "modified": "2020-05-02T08:17:27.000Z",
    "created_by_ref": "identity-f431f809-377b-45e0-aa1c-6a4751cae5ff",
}
```

```

    "name": "CVE-2019-15126",
    "external_references": [
      {
        "source_name": "cve",
        "external_id": "CVE-2019-15126"
      }
    ]
}
]

```

La información pasa a través del *parser* correspondiente definido en 6.2.2 y se mapea al lenguaje OWL para que también pueda ser procesada por el razonador posteriormente. En la Figura 31 se observa la creación de una nueva instancia de tipo *Vulnerability* con la información procedente de fuentes de *Threat Intelligence*. Esta vulnerabilidad viene definida por su nombre (*name*: CVE-2020-15126), su id, su tipo, su tiempo de creación y/o modificación y la identidad que ha creado dicha vulnerabilidad.

The screenshot shows the Protégé ontology editor interface. The top navigation bar includes tabs for Active ontology, Entities, Classes, Object properties, Data properties, Individuals by class, DL/Lviz, Annotations, Usage, and OntoGraf. The main window displays the Class hierarchy under the 'Vulnerability' class. A search bar at the top right shows 'Individuals: CVE-2019-15126'. The left sidebar lists various ontology classes such as Marking\_Definition, Note, Relationship, Report, Risk, Risk\_Assessment, Risk\_Impact, Risk\_Management, Risk\_Owner, Risk\_Probability, Risk\_Severity, Safeguards, Security\_Events, Sighting, Threat, Threat\_Actor, Tool, and Vulnerability. The central workspace shows the asserted individual 'CVE-2019-15126' highlighted in blue. Below it, the 'Description' section contains the string 'CVE-2019-15126'. To the right, the 'Property assertions' section shows several asserted properties for the individual, including 'created' (xsd:dateTimeStamp), 'created\_by\_ref' (identity-f431f809-377b-45e0-aa1c-6a4751cae5ff), 'type' (Vulnerability), and 'spec\_version' (xsd:string). The bottom right corner shows a status bar with '7 8 9 0'.

Figura 31. Prueba de validación correspondiente a la Integración de información STIX sobre una Vulnerabilidad.

Active ontology | Entities | Classes | Object properties | Data properties | Individuals by class | OWLviz | DL Query | SWRLTab x

Annotations | Usage | Individuals |

Class hierarchy: Campaign

owl:Thing (owl:Thing)

- Asset
- Asset Valuation
- Attack\_Pattern
- Bundle
- Campaign

Context

- Correlated\_Anomaly
- Course\_of\_Action
- Cyber\_Observables
- Detected\_Anomaly
- Dictionary
- Event
- Grouping
- Identity
- Incident
- Indicator
- Infrastructure
- Intrusion\_Set
- Kill\_Chain\_Phase
- Location

Malware

- Malware\_Analysis
- Marking\_Definition
- Namenazas
- Note
- Observed\_Data
- Opinion
- Relationship
- Source

Direct instances: c94dd9ad\_c437\_4cf5\_a1e1\_2626165c38f

For: Campaign

'Green Group Attacks Against Finance'

Individuals: 'Green Group Attacks Against Finance'

Evil Org

Green Group Attacks Against Finance

- 027e31c.e748\_480d\_b4e6\_ce3bcab3fb
- 0871dec4\_47ad\_4180\_8b75\_4b5a5a7eb810
- 154a384d\_6620\_47f1\_ba37\_0fed69c46ea
- 16abbfd7\_1105\_481f\_ab74\_8f5fbaa2a1
- 189f933c\_37e0\_47f0\_ba37\_0fed69c46ea
- 1ba4fc5\_3c\_47f6\_beec\_8a5138f240f1
- 25e41389\_2e42\_4b74\_940e\_7415ad194fc
- 27caed9a\_ae5e\_af4f\_920c\_a6ca2d91814fc
- 282b0bd9\_67ef\_4x4f\_8679\_fa223c2779
- 2cc0df13\_644b\_4b01\_90dd\_f773205bf40
- 43b751ba\_a0f1\_41a7\_80a4\_222ab97297c
- 4fc037b5\_fa50\_4b67\_f819\_212dc61638f9
- 51bb82ed\_6e36\_4db8\_8d29\_5f69ce538a8
- 56ba7809\_0174\_4bea\_8886\_f62d2ad073c
- Sae63417\_b903\_49ef\_9538\_621b2e8197f
- Se6d1c64\_4fa3\_431a\_bb2d\_54380215a27
- 6c19c3b2\_6291\_47bf\_h8a7\_6606c0b5a0
- G990d7d5\_3f08\_4ba0\_9ea9\_7f4c7a3b2d20
- G99d06e6\_9202\_485c\_be36\_c051875c169c

Description: 'Green Group Attacks Against Finance'

Property assertions: 'Green Group Attacks Against Finance'

Aliases some stixlist

Attributed to some Intrusion\_Set

Attributed to some Threat\_Actor

Campaign

Compromises some Infrastructure

Created some xsddateTimeStamp

Created\_by\_ref some Identity

Description some xsd:string

First\_seen some xsddateTimeStamp

IsIndicatively some Indicator

Last\_seen some xsddateTimeStamp

Modified some xsddateTimeStamp

Name only xsd:string

Object\_marking\_refs some Marking\_Definition

Objective some xsd:string

Originates from some Location

owl:Thing or (objectRefs some Marking\_Definition)

Object property assertions

- created\_by\_ref identity-->f41f809-377b-45e0-a1c-6a4751cae5ff

Data property assertions

- name "Green Group Attacks Against Finance"^^xsd:string
- id "campaign--8e2ed2b-17d4-4cbf-938f-98ee46b3cd3"^^xsd:string
- description "Campaign by Green Group against a series of targets in the financial services sector."^^xsd:string
- modified "2016-04-06T20:03:00.000Z"^^xsddateTimeStamp
- created "2016-04-06T20:03:00.000Z"^^xsddateTimeStamp

Negative object property assertions

Negative data property assertions

**Figura 32. Prueba de validación correspondiente a la Integración de información STIX sobre una Campaña**

Se verifica que la información de *Threat Intelligence* en formato STIX se integra correctamente en la ontología.

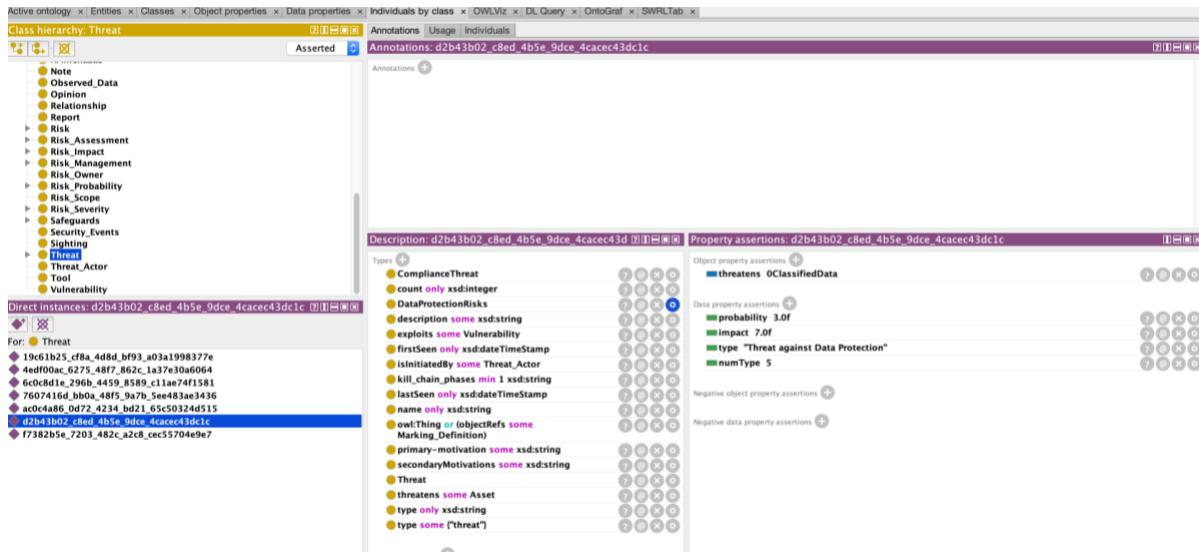
## 7.4 CREACIÓN DE AMENAZAS Y RIESGOS

En tercer lugar, se comprueba que la información es procesada por el sistema y se generan amenazas y riesgos conforme a lo establecido en las reglas SWRL.

Por un lado, van a existir amenazas que derivan de las dependencias entre los activos. Este es el caso de amenazas de Cumplimiento de Protección de Datos. De estas, se derivan Riesgos de Cumplimiento de Protección de Datos. Las instancias o ejemplares de estas amenazas y riesgos se crean a partir de las reglas llamadas ‘Reglas de inventario de Amenazas’ y ‘Reglas de inventario de Riesgos’. En este caso, la regla para este riesgo es la que se observa en la siguiente imagen.

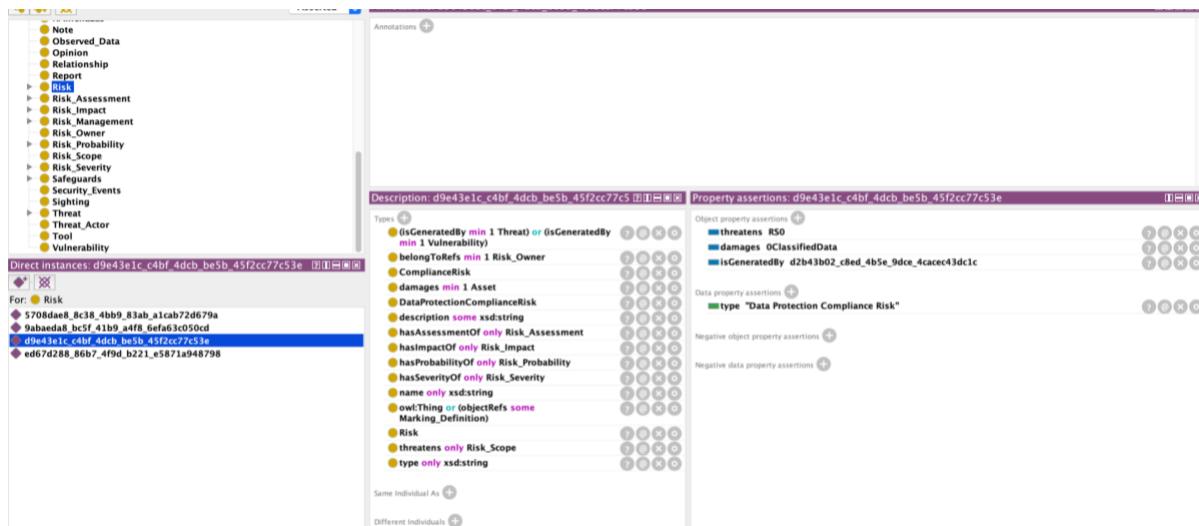
**Figura 33. Prueba de validación correspondiente a la Definición de Reglas SWRL sobre la creación de Riesgos de Cumplimiento de Protección de Datos**

Las instancias de amenazas y riesgos que se crean a partir de estas reglas son las que se observan en las siguientes imágenes.



This screenshot shows the 'Active ontology' interface with the 'Classes' tab selected. A specific class, 'Threat', is highlighted in yellow. Below the class hierarchy, a list of direct instances is shown, including one named 'd2b43b02\_c8ed\_4b5e\_9dce\_4cacec43dc1c'. The right side of the interface displays the properties and assertions for this individual. Under 'Description' and 'Property assertions', several properties are listed with their values: 'threatens' is asserted to 'OclassifiedData' (with probability 3.0f, impact 7.0f, type "Threat against Data Protection", and numType 5). Other properties like 'countOnly xsd:integer', 'exploits some Vulnerability', and 'name only xsd:string' are also listed. The interface includes tabs for Annotations, Usage, and Individuals, and various toolbars at the top.

Figura 34. Prueba de validación correspondiente a la creación de una Amenaza a partir de los Activos.



This screenshot shows the 'Active ontology' interface with the 'Classes' tab selected. A specific class, 'Risk', is highlighted in yellow. Below the class hierarchy, a list of direct instances is shown, including one named 'd9e43e1c\_c4bf\_4dc8\_be5b\_45f2cc77c53e'. The right side of the interface displays the properties and assertions for this individual. Under 'Description' and 'Property assertions', several properties are listed with their values: 'isGeneratedBy' is asserted to 'min 1 Threat' or 'isGeneratedBy min 1 Vulnerability' (with damages R50, OclassifiedData, and isGeneratedBy d2b43b02\_c8ed\_4b5e\_9dce\_4cacec43dc1c). Other properties like 'belongToRefs min 1 Risk\_Owner', 'ComplianceRisk', 'damages min 1 Asset', 'DataProtectionComplianceRisk', 'hasAssessmentOf only Risk\_Assessment', 'hasImpactOf only Risk\_Impact', 'hasProbabilityOf only Risk\_Probability', 'hasSeverityOf only Risk\_Severity', and 'name only xsd:string' are also listed. The interface includes tabs for Annotations, Usage, and Individuals, and various toolbars at the top.

Figura 35. Prueba de validación correspondiente a la creación de un Riesgo de tipo Data Protection Compliance

Por otro lado, según lo definido en las reglas SWRL ‘Reglas de Anomalías’, una anomalía genera una amenaza por sí misma si supera un umbral. En este caso, el umbral se estableció en un valor de 4. Si el valor de *suspicious\_value* de una instancia de anomalía supera este valor, significa que existen anomalías similares con características y propiedades similares y que, por tanto, son susceptibles de generar una amenaza en el sistema.

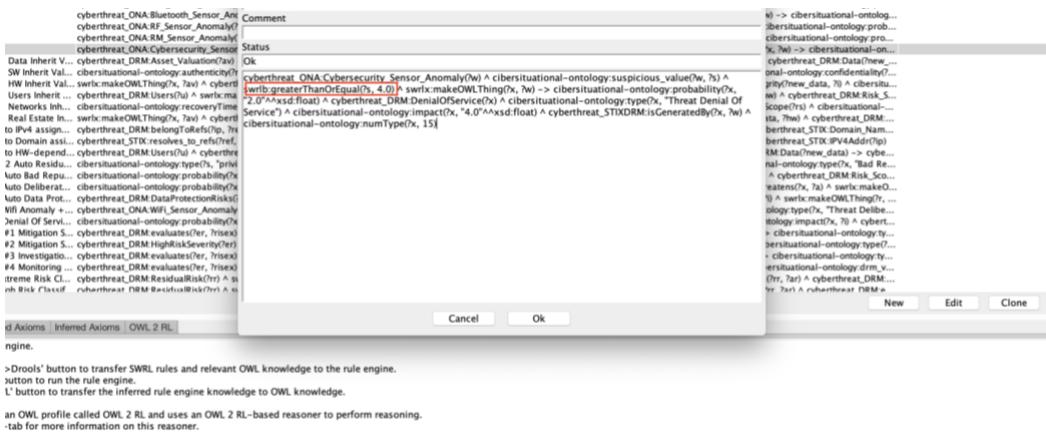


Figura 36. Prueba de validación correspondiente a la Definición de Reglas SWRL para Anomalías

Por tanto, siguiendo el ejemplo de las anomalías creadas en la Figura 29 y Figura 30, la anomalía tipo Wifi no genera ninguna amenaza, pero la anomalía tipo Ciberseguridad sí. Y, en este caso, será una amenaza de tipo *Denial of Service*, según lo dispuesto en la definición de la regla. Esto significa que existen varias anomalías tipo Ciberseguridad que proceden del mismo dispositivo.

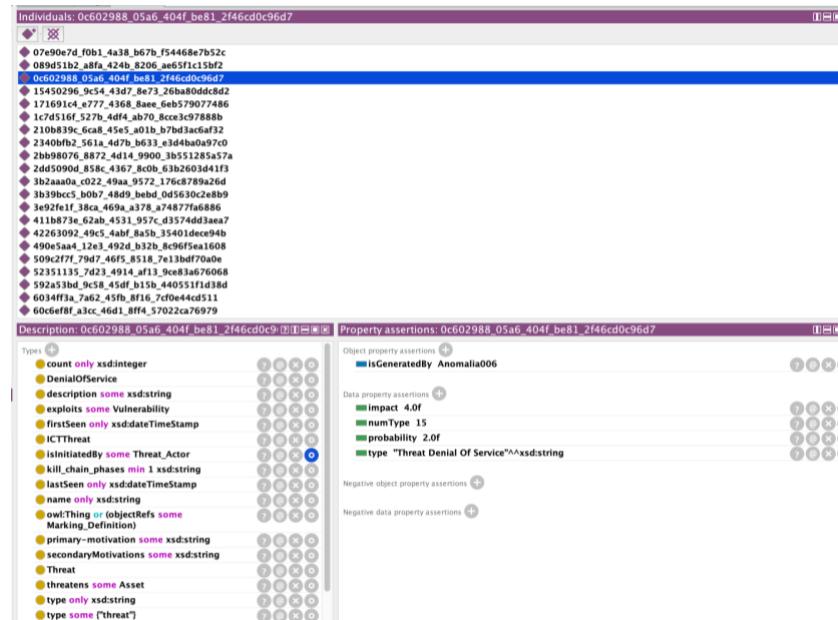


Figura 37. Prueba de validación correspondiente a la creación de una Amenaza a partir de una Anomalía.

Por otro lado, la información STIX también es susceptible de generar amenazas conforme a las reglas. En este caso, se genera una amenaza si existe un activo de tipo software que tiene una relación de dependencia con datos clasificados (*Classified Data*).

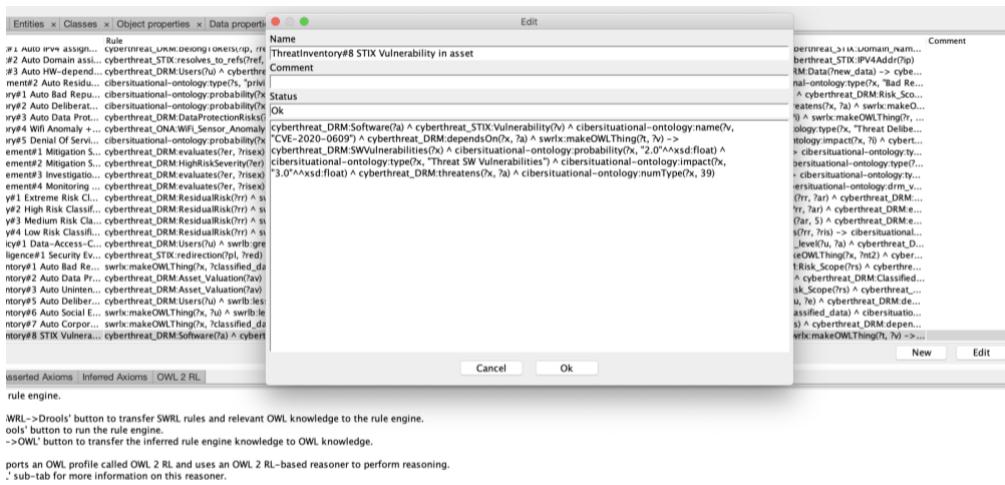


Figura 38. Prueba de validación correspondiente a la Definición de Reglas SWRL para información STIX

La amenaza que se generaría si existiese un activo que cumpliese esas características, sería la presentada en la siguiente figura.

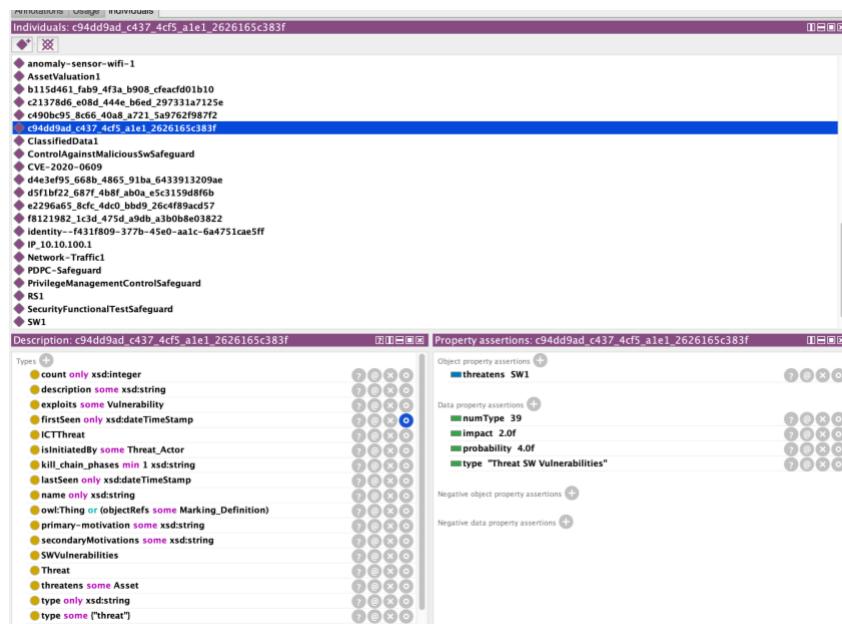


Figura 39. Prueba de validación correspondiente a la creación de una Amenaza a partir de información STIX

Como resultado de las amenazas se crean riesgos, que posteriormente serán analizados y evaluados. Esto es realizado automáticamente a partir de las reglas SWRL ‘Reglas de Evaluación de Riesgos’, creadas para este trabajo y en común con las reglas ya creadas en [32]. El riesgo se caracteriza por tener un impacto y una probabilidad. Para establecer dichos valores se realiza un análisis cualitativo de manera que el rango de valores oscila entre 0, despreciable, y 10, inaceptable.

En la Figura 40, se observa la creación de un riesgo como consecuencia de las anomalías de tipo Ciberseguridad.

Figura 40. Prueba de validación correspondiente a la creación de un Riesgo de tipo Denial of Service

Los riesgos se evalúan conforme a un impacto y una probabilidad. Se realiza un análisis potencial y residual del riesgo. En el caso del análisis potencial muestran las consecuencias que tendría sobre la organización la materialización de una determinada amenaza. El análisis residual tiene en cuenta la existencia de salvaguardas como medidas para enfrentarse a esas amenazas y reducir el impacto y la probabilidad de que ocurran los riesgos.

Figura 41. Prueba de validación correspondiente a la creación de un Riesgo Potencial

En la imagen anterior se puede observar el riesgo potencial correspondiente a la evaluación del riesgo de la Figura 40. Este riesgo se crea automáticamente con las reglas SWRL, se calcula siguiendo la ecuación (1) y el resultado se refleja en el parámetro *potentialRisk*.

Una vez evaluada la existencia de este riesgo potencial se analiza la existencia de salvaguardas que se enfrenten a dicho riesgo. Como se observa en el valor *isMitigatedBy* existe una salvaguarda de tipo *Security Functional Test Control* que pretende reducir la probabilidad y el impacto en caso de que se materialice el riesgo. Como consecuencia de la aplicación de la salvaguarda, se genera un riesgo residual. El valor de dicho riesgo se calcula automáticamente con la ecuación (2) y se refleja en el parámetro *actualRisk*.

En la siguiente imagen, se puede observar una instancia de la clase Salvaguardas de tipo *Security Functional Test Control*.

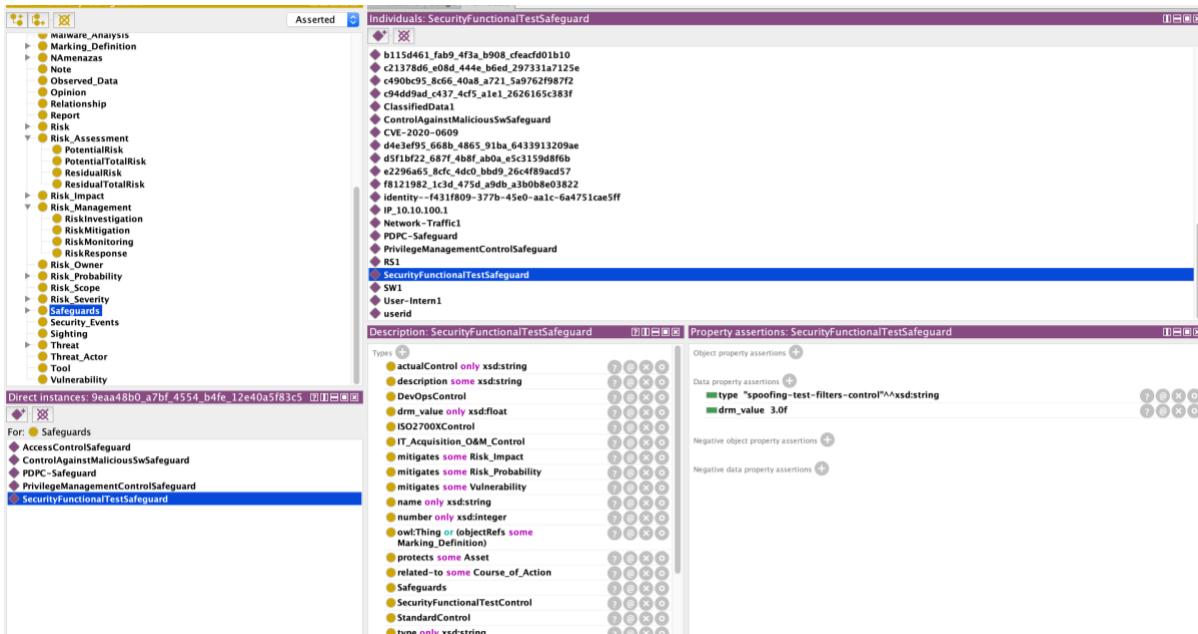


Figura 42. Instancia de tipo Salvaguarda

De esta manera, se verifica el correcto funcionamiento de las siguientes partes:

- Generación de amenazas y riesgos a partir de información de activos.
- Generación de amenazas y riesgos a partir de información de anomalías.
- Generación de amenazas y riesgos a partir de información de STIX.
- Evaluación de riesgos en potenciales y residuales

## 7.5 CÁLCULO DEL RIESGO

En este apartado se verifica el correcto cálculo del riesgo dinámicamente según lo explicado en el apartado 6.3 y se genera correctamente un historial de datos como el que se observa a continuación, en formato JSON:

```

    "Time":"2020-06-14T13:21:21.470Z",
    "Potential Total Risk":"6.8461537",
    "Residual Total Risk":"3.5384614",
    "Potential Total Risk Continuous":"6.240740855341922",
    "Residual Total Risk Continuous":"2.944444494992657",
    "Threat Total Number":"13.0",
    "Risks": [
        {
            "Risk Name":"Denial of Service Risk",
            "Potential Risk":"8.0",
            "Residual Risk":"5.0",
            "Threat Number":"6.0",
            "Impact Value":"5.0",
            "Probability Value":"3.0"
        },
        {
            "Risk Name":"Deliberated Unauthorized Access Risk",
            "Potential Risk":"6.0",
            "Residual Risk":"2.0",
            "Threat Number":"5.0",
            "Impact Value":"4.0",
            "Probability Value":"2.0"
        },
        {
            "Risk Name":"Bad Reputation Risk",
            "Potential Risk":"5.0",
            "Residual Risk":"3.0",
            "Threat Number":"1.0",
            "Impact Value":"2.0",
            "Probability Value":"3.0"
        },
        {
            "Risk Name":"Data Protection Compliance Risk",
            "Potential Risk":"6.0",
            "Residual Risk":"3.0",
            "Threat Number":"1.0",
            "Impact Value":"3.0",
            "Probability Value":"3.0"
        }
    ],
    "Strategies": [
        {
            "Risk":"Bad Reputation Risk",
            "Recommendation Strategy":"Risk Monitoring Strategy",
            "Risk Value":"3.0"
        },
        {
            "Risk":"Denial of Service Risk",
            "Recommendation Strategy":"Risk Investigation Strategy",
            "Risk Value":"5.0"
        },
        {
            "Risk":"Deliberated Unauthorized Access Risk",
            "Recommendation Strategy":"Risk Monitoring Strategy",
            "Risk Value":"2.0"
        },
        {
            "Risk":"Data Protection Compliance Risk",
            "Recommendation Strategy":"Risk Monitoring Strategy",
            "Risk Value":"3.0"
        }
    ]
}

```

**Figura 43. Historial de datos resultantes del cálculo del riesgo**

## 7.6 INTERFAZ DE VISUALIZACIÓN

Por último, se va a verificar que se observan los resultados obtenidos por el sistema correctamente.

La interfaz de visualización simplifica la visualización de los datos obtenidos. Para su realización se ha utilizado la librería de Java JFreeChart.

Para acceder a cada una de las opciones existe un menú principal que ofrece dos opciones: visualizar datos referentes al riesgo total de la organización o visualizar datos en función del tipo de riesgo. La primera opción incluye:

- Evolución de los Riesgos Potenciales y Residuales Finales Totales de la organización.
- El Riesgo Potencial y Residual Total Instantáneo de la organización.

La segunda opción incluye:

- Los Riesgos Potenciales y Residuales de cada uno de los tipos existentes en la organización.
- Un mapa de riesgos.

La siguiente figura muestra la Evolución del riesgo potencial total y el riesgo residual total de la organización. Estos resultados son no discretos por lo que tienen en cuenta los riesgos anteriores. En este caso se han insertado 25 anomalías y 3 datos de STIX.

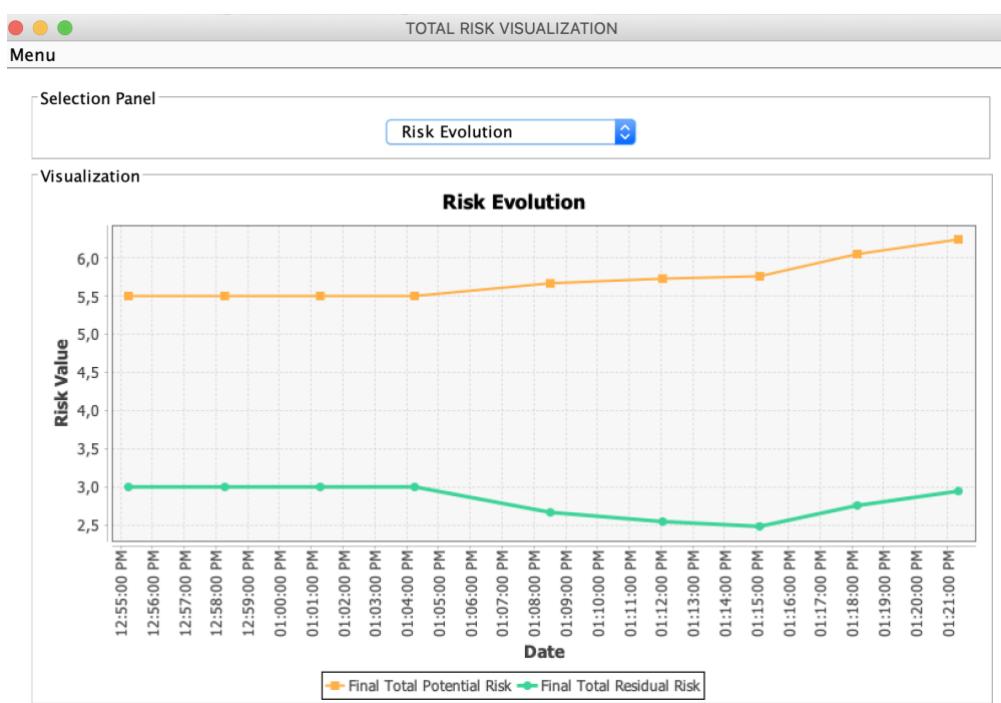


Figura 44. Interfaz de Visualización: Riesgos Totales Finales Potenciales y Residuales

Para visualizar el Riesgo Instantáneo Total de la organización basta con elegir la opción *Instantaneous Total Risk*. Esta opción muestra una gráfica con el riesgo total presente en la organización en una escala del 0 al 10, donde 0 es despreciable y 10, inaceptable. Se compara estos resultados con el valor máximo que puede tener el riesgo, que es 10.

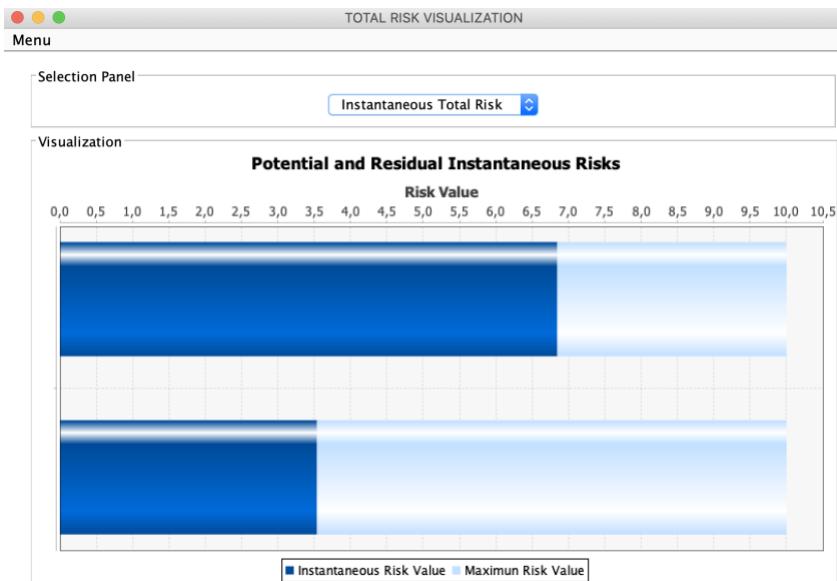


Figura 45. Interfaz de Visualización: Riesgo Instantáneo Total Potencial y Residual

La interfaz permite ir más allá de los riesgos totales para poder visualizar los riesgos potenciales y residuales de cada uno de los riesgos existentes en la organización en ese instante.

En este caso, existen 4 tipos de riesgos diferentes. Para todos se dispone de salvaguardas y medidas de control, puesto que existen riesgos tanto potenciales como residuales para cada uno.



Figura 46. Interfaz de Visualización: Riesgo Potencial y Residual por Tipo de Riesgo

La visualización también incluye el soporte a la toma de decisiones para facilitar que la propuesta de estrategias sea lo más dinámica para el usuario. Para ello, se ha generado un mapa de riesgos donde se observan los riesgos en función de su impacto y probabilidad. De esta manera, se facilita la evaluación de los riesgos, la decisión sobre cuáles se van a tratar y la propuesta de procesos de actuación para cada uno de los riesgos.

En la siguiente imagen se observa el mapa de riesgos derivado del gráfico anterior.

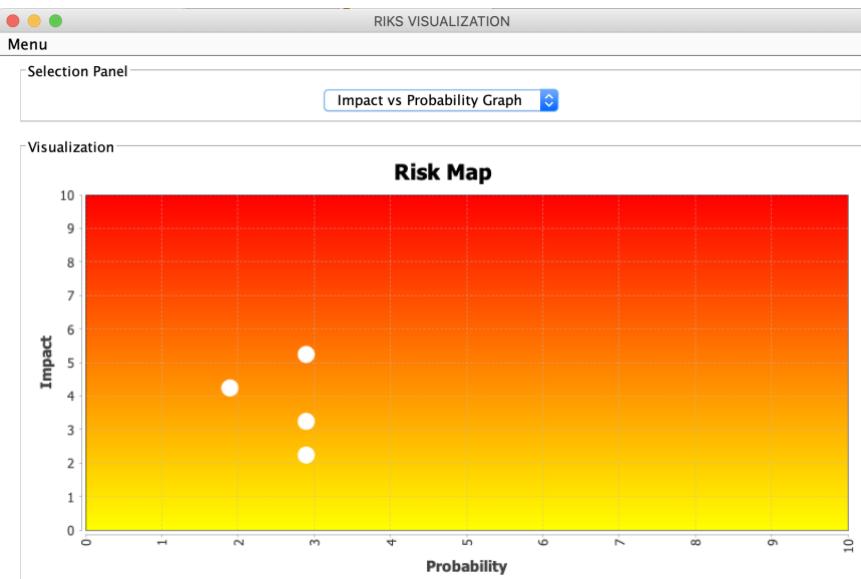


Figura 47. Interfaz de Visualización: Mapa de Riesgos

Por último, si se selecciona cada uno de los riesgos, se proponen estrategias de actuación para el soporte a la toma de decisiones.

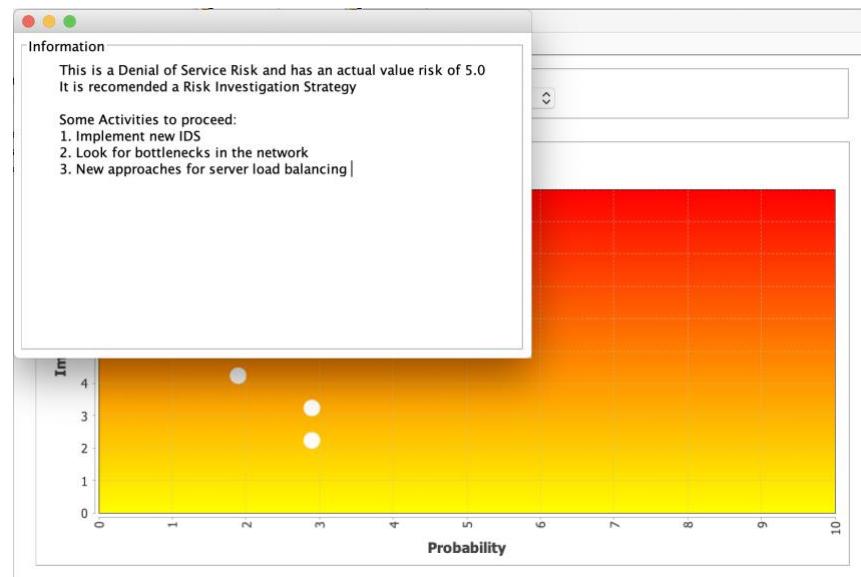


Figura 48. Interfaz de Visualización: Soporte a la toma de decisiones

## 7.7 CASO PRÁCTICO

En este apartado se pretende mostrar la funcionalidad práctica del sistema implementado.

Para llevar a cabo esta demostración se ha tenido en cuenta un caso real. En 2019 se descubrió una vulnerabilidad que afectaba a millones de dispositivos con Wifi, *smartphones*, *tablets*, portátiles, etc. Esta vulnerabilidad permitía al atacante interceptar y descifrar los paquetes de red enviados de

forma inalámbrica en redes Wifi, basadas en WPA. Esto se producía en los momentos de disociación durante el estado de transición del Wifi, entre el dispositivo y el punto de acceso. En ese momento, los atacantes encriptan la información haciendo uso de la clave puesta a cero para extraer información. De esta manera, pueden darse fugas de información y accesos no autorizados que afectan a la confidencialidad del sistema.

Se trata de la vulnerabilidad Kr00K. Esta se encuentra identificada como una CVE, en concreto CVE-2019-15126 y fue publicada por la compañía ESET [36].

En el caso que aquí se va a exemplificar, se muestra la funcionalidad del sistema desarrollado en caso de que se dispusiera de un dispositivo susceptible de verse afectado por esa vulnerabilidad.

El escenario propuesto es el siguiente. Se trata de un caso concreto dentro de una organización donde existe un empleado que tiene acceso a datos clasificados y que hace uso de un ordenador conectado a una red WiFi. El punto de acceso a esta red es un router que está afectado por la vulnerabilidad Kr00k.

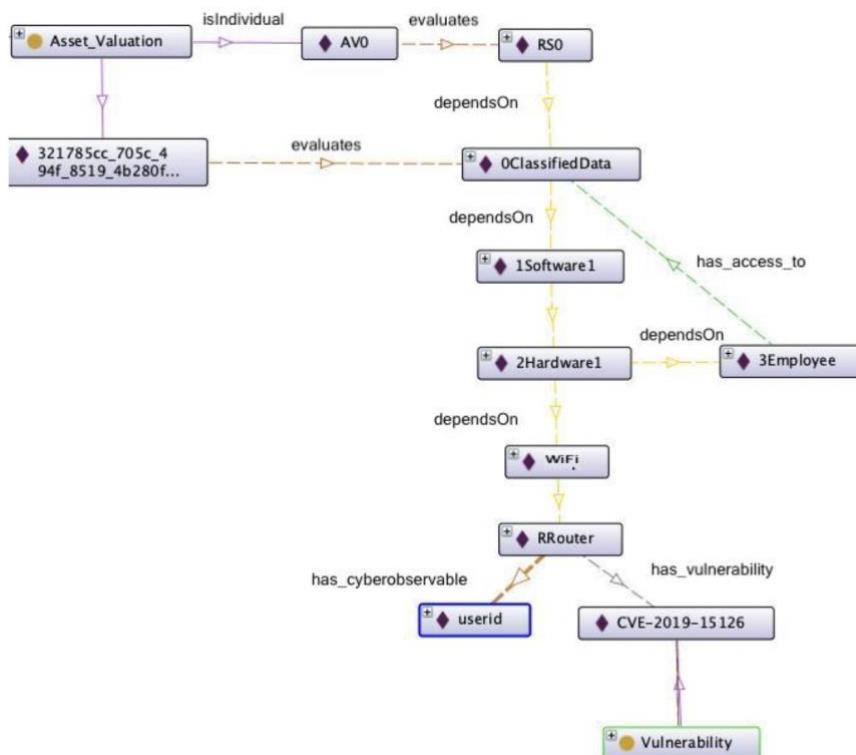


Figura 49. Escenario del ejemplo práctico

Una vulnerabilidad puede ser explotada por un atacante para causar daño. Puesto que la información STIX incluye información relacionada con CVEs, la organización es capaz de integrar la información de la vulnerabilidad.

Por otro lado, la existencia de una anomalía de tipo WiFi permite detectar de manera temprana una actividad anómala en la red wifi en la que se encuentra el dispositivo afectado por la vulnerabilidad.

La ejecución del sistema sigue el siguiente flujo.

En primer lugar, se integran los datos sobre activos en la ontología DRM. Cuando se detecta la llegada de información de inteligencia de amenazas en formato STIX se introduce en la ontología CTI. De la misma manera, cuando se detecta información relacionada con anomalías de tipo WiFi se introduce en la ontología ONA.

A continuación, se ejecutan las reglas. Se ejecutan las ‘Reglas de Inventory de Amenazas’ y las ‘Reglas de Inventory de Riesgos’ que afectan a los activos y generan las amenazas y riesgos a los que están expuestos. Como consecuencia de la existencia de riesgos, se ejecutan las ‘Reglas de Evaluación de Riesgos’ para calcular los riesgos potenciales y residuales.

Estos resultados pasan por el subsistema de cálculo del riesgo para su cómputo.

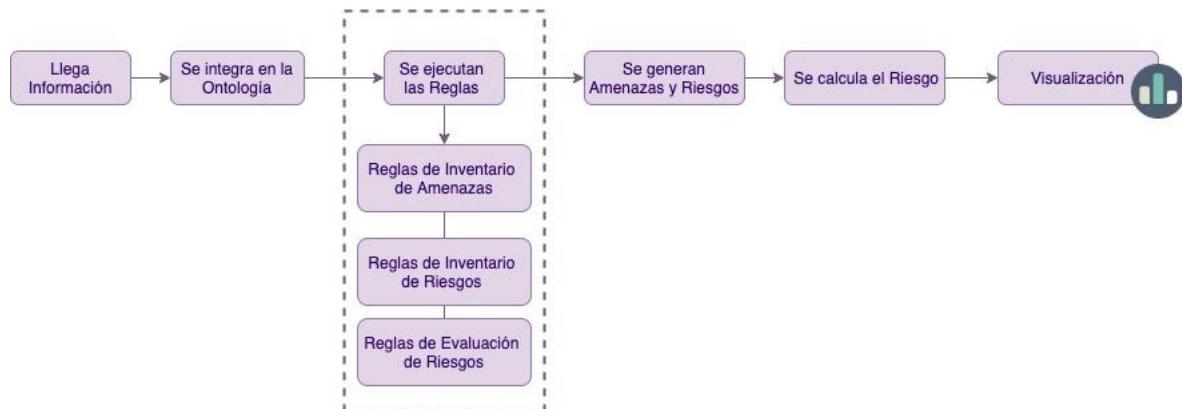


Figura 50. Flujo de la ejecución del ejemplo práctico

Los resultados de la ejecución del sistema en un escenario como este, generan el siguiente conocimiento que se representa en la Figura 51.

Por un lado, se genera una amenaza que representa la posibilidad de que se produzcan fugas de información (*Deliberated Information Leak Threat*). Como consecuencia se genera un riesgo y se realiza el cálculo del riesgo dinámico. El riesgo se evalúa y se gestiona.

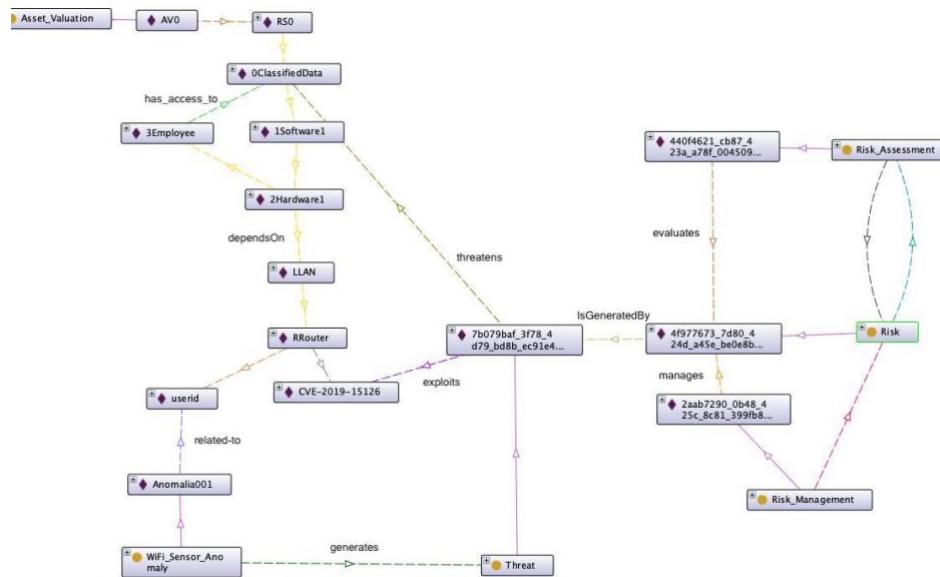


Figura 51. Resultado del ejemplo práctico

El resultado se puede observar gráficamente mediante la interfaz de visualización. Aparecen otros riesgos también porque son los que afectan a los datos clasificados como el riesgo de Mala reputación y el de Incumplimiento de Protección de Datos.

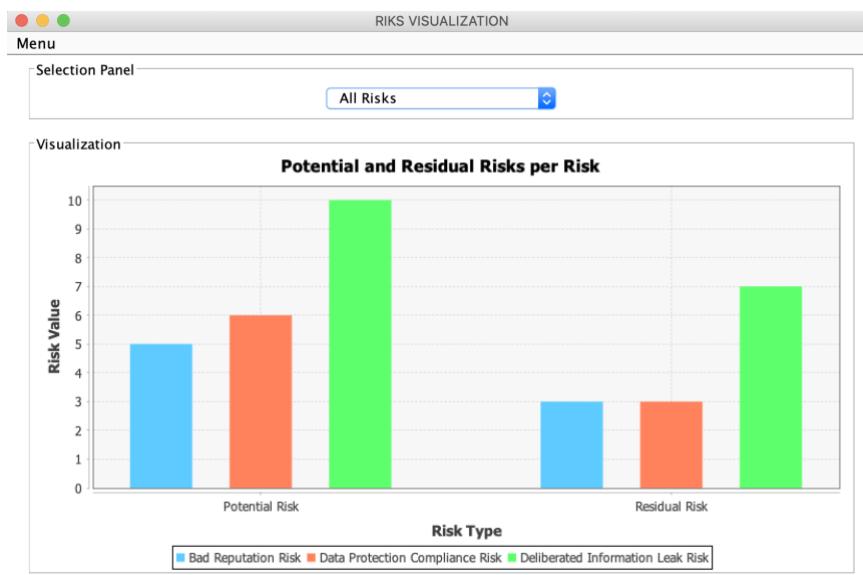


Figura 52. Riesgos resultantes del ejemplo práctico en el sistema de visualización

Por último, se proponen posibles estrategias que puedan llevarse a cabo para disminuir el impacto producido por el riesgo.

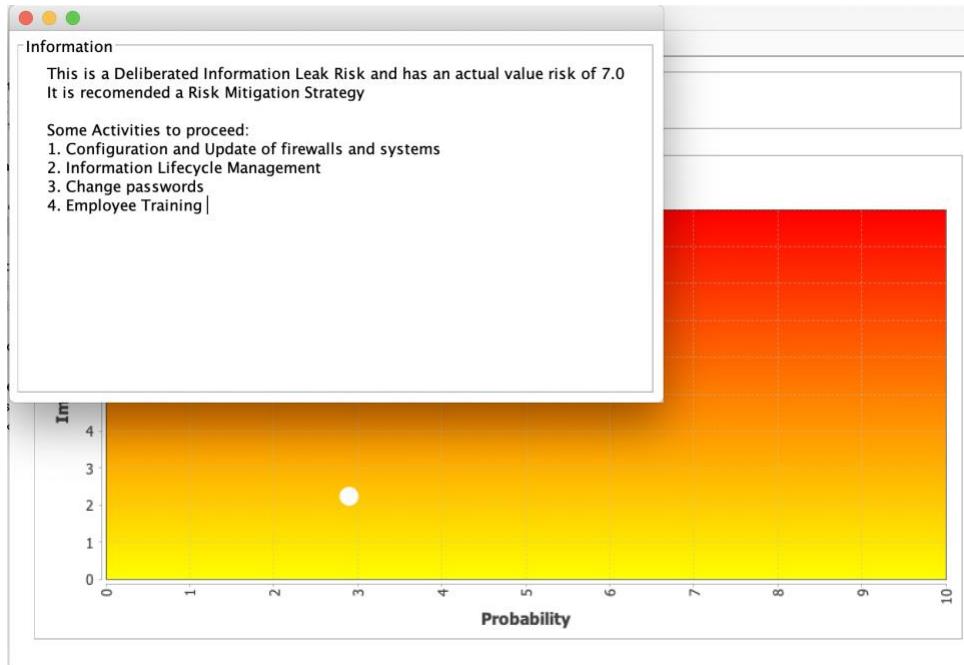


Figura 53. Soporte a la toma de decisiones del ejemplo práctico

La contribución del sistema implementado es que permite detectar la existencia de una amenaza derivada de la existencia de esta vulnerabilidad. Para ello, se hace uso de los *parsers* implementados, de las reglas definidas y del razonador. La relación que existe entre las distintas ontologías del sistema

va a permitir detectar posibles amenazas que pueden afectar a la organización. Es posible relacionar anomalías, con datos sobre inteligencia de amenazas en formato STIX y datos relacionados con la gestión de riesgos para poder dar una respuesta proactiva, dinámica y en tiempo real o casi real.

## 8 CONCLUSIONES

---

### 8.1 CONCLUSIONES

La presente línea de trabajo se ha centrado en el desarrollo de un entorno basado en las ontologías que permiten integrar en una estructura unificada datos heterogéneos de diversas fuentes de información. El entorno de estudio se enmarca en ciberseguridad donde la existencia de múltiples datos es cada vez más frecuente.

Para ello, en primer lugar, se ha estudiado el marco actual de la ciberseguridad donde se ha analizado la seguridad en las redes de telecomunicación y la existencia de tecnologías relacionadas como pueden ser herramientas de gestión de vulnerabilidades, de gestión de activos, de análisis de riesgos, etc. Por otra parte, se ha analizado la necesidad de compartir información sobre amenazas entre las organizaciones, lo cual conlleva la estandarización de los lenguajes utilizados para el intercambio de dicha información. Se ha concluido la necesidad de unificar todos estos datos en una estructura común, la cual permita relacionarlos para identificar amenazas y riesgos de la manera más temprana posible.

Para el desarrollo de esa estructura unificada, se ha propuesto el uso de ontologías. Se han analizado los distintos lenguajes de definición de ontologías existentes actualmente, así como los lenguajes de definición de reglas de comportamiento y los razonadores semánticos. Se ha realizado una comparación exhaustiva y se ha concluido el uso del lenguaje OWL para la definición de la ontología, el uso del lenguaje de reglas SWRL y del razonador Pellet como motor de inferencia por ser los más usados en la Web Semántica y los de mayor compatibilidad con las tecnologías en el ámbito de las ontologías. El uso de las ontologías aporta coherencia semántica, consistencia e inteligencia al sistema. Con ellas, se consigue realizar una representación formal de la información que puede ser entendida semánticamente por máquinas y procesada para la extracción de nuevo conocimiento.

Una vez identificados los recursos para la implementación del sistema se definen los requisitos del sistema y su arquitectura. Para facilitar el desarrollo de la ontología se ha utilizado el editor Protégé que ofrece una sencilla interfaz.

Otro objetivo propuesto para este trabajo era el cálculo del riesgo partir de la información inferida en el sistema de ontologías. Para ello, se ha desarrollado un programa en Eclipse utilizando OWL API. Los resultados de este cálculo son los riesgos potenciales y residuales de la organización y propuestas de estrategias para el soporte a la toma de decisiones. Para poder visualizar los datos de una manera más dinámica, se ha realizado una interfaz de visualización mediante la librería JFreeChart de Java.

Finalmente, se ha validado el funcionamiento del sistema. Se han introducido una serie de datos de fuentes diversas. Estos datos son información relacionada con anomalías, inteligencia de amenazas y activos y tienen distinta sintaxis y formato. Se ha comprobado que los datos se introducen correctamente en el sistema de anomalías y se crean las amenazas y riesgos correspondientes. Se ha verificado que el sistema del cálculo del riesgo realiza el cómputo adecuadamente y se visualiza correctamente en la interfaz gráfica desarrollada.

El valor de este trabajo reside en la consolidación de distintos recursos y herramientas en un sistema. Gracias a la implementación de este sistema integrado se consigue gestionar un entorno heterogéneo y elaborar un mismo mecanismo de acceso a la información. De esta manera, se ofrece una visión global del estado y compromiso del sistema y sirve para dar soporte a la conciencia cibersituacional. Además, para el desarrollo del sistema se ha hecho uso de herramientas de software libre y tecnologías actuales, gracias a las cuales pueden realizarse en el futuro pruebas y mejoras de su funcionamiento.

## 8.2 LÍNEAS FUTURAS

Durante el desarrollo del trabajo se han identificado nuevas funcionalidades que podrían implementarse en el futuro para enriquecer el sistema.

- La ontología desarrollada puede ser mejorada y enriquecida con nueva información procedente de futuras versiones del formato STIX, o con nueva información relacionada con anomalías (como, por ejemplo, anomalías de correlación).
- La información procedente de las fuentes externas puede ser modificada para adaptarla a nuevos módulos en el sistema.
- En el futuro puede ser necesario añadir nuevas fuentes de información debido a que el sistema es escalable en términos de datos de entrada.
- El sistema se podrá enriquecer con nuevas reglas que prueben otros casos que no se han considerado en el presente trabajo, con el fin de evidenciar nuevos TTP.
- La interfaz de visualización puede ser mejorada para permitir visualizar otro tipo de datos y gráficas mediante su integración con Visual Analytics.

Así mismo, en el desarrollo se han encontrado algunos inconvenientes y se proponen algunas soluciones que podrían desarrollarse en futuras implementaciones.

Dado que la ontología presentada en este trabajo es de grandes dimensiones (la ontología dispone de casi 500 clases y más de 15000 declaraciones de axiomas) la capacidad de cómputo de los sistemas utilizados para su desarrollo se presenta inadecuada para poder llevar a cabo la inferencia de conocimiento en un tiempo reducido. Se considera que, con el gran avance en la tecnología y su rápida evolución, los sistemas podrán hacer frente a las velocidades y capacidades requeridas para mejorar la eficiencia y el funcionamiento del sistema.

Por otro lado, los razonadores semánticos actuales tienen una capacidad insuficiente para ontologías de grandes dimensiones, con las que trabajan de manera más lenta e inefficiente. Sin embargo, existen estudios [37] que hablan del desarrollo de nuevos razonadores que son capaces de cargar ontologías extensas de manera rápida y eficaz. Esto supone un nuevo paso para mejorar el desarrollo de los razonadores semánticos actuales con el fin de hacer que el proceso de razonamiento sea escalable a ontologías de grandes dimensiones.

En suma, el sistema desarrollado es un sistema que puede ser adaptado a muchos entornos en el ámbito de la ciberseguridad. Este sistema presenta un entorno de lo que se podría realizar en un escenario real de detección de amenazas y evaluación de riesgos. Además, presenta un amplio margen de mejoras y que debido al uso de tecnología de código libre se presenta flexibilidad y facilidad para llevarlas a cabo.

## 9 REFERENCIAS

---

- [1] M. S. Jalali, J. P. Kaiser, M. Siegel, y S. Madnick, «The Internet of Things Promises New Benefits and Risks: A Systematic Analysis of Adoption Dynamics of IoT Products», *IEEE Secur. Priv.*, vol. 17, n.º 2, pp. 39-48, mar. 2019, doi: 10.1109/MSEC.2018.2888780.
- [2] «CSIRT-CV e INTECO-CERT publican el informe: “Deteción de APTs”», *INCIBE*, may 20, 2013. <https://www.incibe.es/protege-tu-empresa/blog/deteccion-apts> (accedido jun. 13, 2020).
- [3] «itds16.pdf». Accedido: jun. 13, 2020. [En línea]. Disponible en: <https://www.itdigitalsecurity.es/whitepapers/content-download/09948bb9-a31e-4135-9e28-88f28053ae5a/itds16.pdf?s=ra>.
- [4] «PAe - MAGERIT v.3 : Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información». [https://administracionelectronica.gob.es/pae\\_Home/pae\\_Documentacion/pae\\_Metodolog/pae\\_Magerit.html](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html) (accedido jun. 13, 2020).
- [5] «istr-24-2019-en.pdf». Accedido: jun. 13, 2020. [En línea]. Disponible en: <https://docs.broadcom.com/doc/istr-24-2019-en>.
- [6] R. Brown y R. M. Lee, «The Evolution of Cyber Threat Intelligence (CTI): 2019 SANS CTI Survey», p. 16, 2019.
- [7] A. Fielder, «Modelling the Impact of Threat Intelligence on Advanced Persistent Threat Using Games», en *From Lambda Calculus to Cybersecurity Through Program Analysis: Essays Dedicated to Chris Hankin on the Occasion of His Retirement*, A. Di Pierro, P. Malacaria, y R. Nagarajan, Eds. Cham: Springer International Publishing, 2020, pp. 216-232.
- [8] «Escáner de vulnerabilidades Nessus Essentials | Tenable®». <https://es-la.tenable.com/products/nessus/nessus-essentials> (accedido may 02, 2020).
- [9] «OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner». <https://www.openvas.org/> (accedido may 02, 2020).
- [10] «PILAR». <https://www.ccn-cert.cni.es/soluciones-seguridad/ear-pilar.html> (accedido may 26, 2020).
- [11] «About CybOX (Archive) | CybOX Project Documentation». <https://cyboxproject.github.io/about/> (accedido mar. 04, 2020).
- [12] «Introduction to STIX». <https://oasis-open.github.io/cti-documentation/stix/intro> (accedido mar. 04, 2020).
- [13] «Introduction to TAXII». <https://oasis-open.github.io/cti-documentation/taxii/intro> (accedido mar. 04, 2020).
- [14] «MISP features and functionalities». <https://www.misp-project.org/features.html> (accedido mar. 04, 2020).
- [15] «RDF - Semantic Web Standards». <https://www.w3.org/2001/sw/wiki/RDF> (accedido mar. 04, 2020).
- [16] «RDFS - Semantic Web Standards». <https://www.w3.org/2001/sw/wiki/RDFS> (accedido mar. 04, 2020).
- [17] «OWL - Estándares Web Semánticos». <https://www.w3.org/OWL/> (accedido mar. 04, 2020).
- [18] «OWL 2 Web Ontology Language Document Overview (Second Edition)». <https://www.w3.org/TR/owl2-overview/> (accedido mar. 04, 2020).
- [19] «Jess, the Rule Engine for the Java Platform». <https://www.jessrules.com/jess/> (accedido mar.

04, 2020).

- [20] «RuleML Homepage». <http://ruleml.org/index.html> (accedido abr. 17, 2020).
- [21] «R2ML -- The REWERSE I1 Rule Markup Language | Working Group I1». <http://www.rewerse.net/I1/oxygen.informatik.tu-cottbus.de/rewerse-i1/@q=r2ml.htm> (accedido mar. 04, 2020).
- [22] «SWRL: A Semantic Web Rule Language Combining OWL and RuleML». <https://www.w3.org/Submission/SWRL/> (accedido mar. 04, 2020).
- [23] «protégé». <https://protege.stanford.edu/> (accedido abr. 18, 2020).
- [24] «OWL API». <http://owlapi.sourceforge.net/> (accedido abr. 18, 2020).
- [25] «Apache Jena - Jena Ontology API». <https://jena.apache.org/documentation/ontology/> (accedido abr. 18, 2020).
- [26] «Bossam - Semantic Web Standards». <https://www.w3.org/2001/sw/wiki/Bossam> (accedido mar. 04, 2020).
- [27] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, y Y. Katz, «Pellet: A practical OWL-DL reasoner», *J. Web Semant.*, vol. 5, n.º 2, pp. 51-53, jun. 2007, doi: 10.1016/j.websem.2007.03.004.
- [28] S. Abburu, «A Survey on Ontology Reasoners and Comparison», *Int. J. Comput. Appl.*, vol. 57, p. 7.
- [29] A. Khamparia y B. Pandey, «Comprehensive analysis of semantic web reasoners and tools: a survey», *Educ. Inf. Technol.*, vol. 22, n.º 6, pp. 3121-3145, nov. 2017, doi: 10.1007/s10639-017-9574-5.
- [30] «Fact - Semantic Web Standards». <https://www.w3.org/2001/sw/wiki/Fact> (accedido mar. 04, 2020).
- [31] «Hermit - Semantic Web Standards». <https://www.w3.org/2001/sw/wiki/Hermit> (accedido mar. 04, 2020).
- [32] R. Riesco Granadino, «Contribution to dynamic risk management automation by an ontology-based framework», phd, E.T.S.I. Telecomunicación (UPM), 2019.
- [33] «stix-v2.1-csprd03.pdf». Accedido: may 13, 2020. [En línea]. Disponible en: <https://docs.oasis-open.org/cti/stix/v2.1/csprd03/stix-v2.1-csprd03.pdf>.
- [34] paulagf1396, *paulagf1396/paulagf1396-ontology-owlapi-app*. 2020.
- [35] M. Vega-Barbas, V. Villagra, F. Monje, R. riesco, X. Larriva-Novo, y J. Berrocal, «Ontology-Based System for Dynamic Risk Management in Administrative Domains», *Appl. Sci.*, vol. 9, oct. 2019, doi: 10.3390/app9214547.
- [36] «ESET\_Kr00k.pdf». Accedido: jun. 14, 2020. [En línea]. Disponible en: [https://www.welivesecurity.com/wp-content/uploads/2020/02/ESET\\_Kr00k.pdf](https://www.welivesecurity.com/wp-content/uploads/2020/02/ESET_Kr00k.pdf).
- [37] M. del M. Roldan y J. Aldana Montes, «DBOWL: persistencia y escalabilidad de consultas y razonamientos en la web semántica», ene. 2008.

# ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

## A.1 INTRODUCCIÓN

Este anexo recoge los impactos más relevantes derivados del trabajo desarrollado. Para ello, se realiza un análisis de los aspectos negativos y positivos y de las personas que se puedan ver afectadas por la implantación del proyecto.

En primer lugar, se detalla el contexto en el que se enmarca este trabajo. El presente Trabajo de Fin de Máster se ha desarrollado en el sector tecnológico de los sistemas telemáticos, en concreto en un proyecto que se encuentra en su fase de diseño y desarrollo.

El trabajo forma parte de un proyecto más amplio que consiste en la creación de una plataforma avanzada de conciencia cibersituacional. El proyecto trata de desarrollar un entorno para monitorizar múltiples fuentes de información, procesar los datos recolectados mediante técnicas de aprendizaje automático y sistemas expertos, detectar y predecir patrones avanzados de ataques y calcular el riesgo de la exposición a los mismos.

El desarrollo de este trabajo será beneficioso para el progreso del proyecto en el que se enmarca. Además, permitirá disminuir los impactos negativos derivados de las amenazas que sufren las organizaciones reduciendo los recursos necesarios.

## A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO

En este análisis se recogen los impactos más relevantes que pueden estar relacionados con el proyecto desarrollado.

Teniendo en cuenta el escenario global del Trabajo de Fin de Máster, se pueden clasificar en los siguientes el conjunto de los impactos que se derivan de su aplicación:

### Impactos socioeconómicos

En cuanto a los impactos socioeconómicos se pueden destacar:

- Seguridad y prevención de riesgos.
- Desarrollo empresarial.
- Buen gobierno interno.
- Integración y adaptación social de la innovación.
- Generación de capacidades sobre los trabajadores.
- Impacto económico.

El desarrollo de este TFM asegura la detección de amenazas y riesgos en el entorno de una organización. De la aplicación de este trabajo, se derivan beneficios en la seguridad de los sistemas de una empresa u organización. Por un lado, los sistemas gozarán de mayor seguridad y las brechas de seguridad se reducirán. La probabilidad de que se materialicen amenazas sobre los activos de la organización se ve disminuida debido a la capacidad de respuesta del sistema desarrollado, que se caracteriza por su proactividad.

Por otro lado, el desarrollo empresarial es innegable. La evolución hacia nuevas tecnologías y sistemas de inteligencia conlleva muchas ventajas, entre otras, la mejora de la eficiencia de los

sistemas de seguridad, el aprovisionamiento más rápido de medidas de seguridad y la reducción del uso de recursos.

El análisis de amenazas y riesgos es un proceso que sustenta el buen gobierno de una organización. Con la implantación de este sistema, este proceso será dinámico, ya que va más allá del carácter estático actual, retroalimentando el sistema de seguridad. La prevención de riesgos permite optimizar el uso de recursos disponibles y mejorar las probabilidades de poder cumplir con los objetivos de la organización. Gracias al uso de estándares, además, se comparte información con otros organismos y se beneficia de la cooperación entre empresas con el fin de reducir la incertidumbre que supone estar sujetos a amenazas y riesgos.

Otro aspecto fundamental es que el uso de este sistema supone la automatización y centralización de los sistemas de seguridad. De esta manera, actividades que realizaban seres humanos de manera más costosa, se ejecutan a mayor velocidad y con resultados más eficientes, reduciendo el esfuerzo que implica. Además, el proceso de gestión de riesgos supone un riesgo por error humano, el cual también se reduce. Esta reducción permite que el especialista de seguridad y el analista puedan centrar su actividad profesional en el estudio de los resultados y la ejecución de acciones de manera más rápida ante amenazas que exponen a la organización. Lejos de la eliminación de puestos de trabajo, este sistema supone una transformación para los trabajadores que deberán formarse en las reglas y el razonamiento semántico.

Por otro lado, este proyecto implica un impacto económico positivo puesto que se reducen los efectos negativos que los incidentes provocan sobre los sistemas de una organización y que conlleva uso de recursos económicos para paliar los problemas. Esto supone, por ejemplo, invertir en nuevo material, pagar multas o invertir en recursos para recuperar información confidencial robada. Además, el software que se ha utilizado en la implementación del sistema es gratuito y libre, por lo que se reduce también el coste que supondría el uso de licencias y software privado.

### **Impacto ambiental**

En cuanto a los impactos ambientales se pueden destacar los impactos sobre los recursos materiales y energéticos de la organización. Este trabajo permite la cooperación entre distintos sistemas de protección y seguridad para dar una respuesta común. Ello evita que los sistemas de seguridad actúen de manera individual y generen una respuesta similar. El uso en conjunto de todos los posibles sistemas de seguridad de una organización reduce el consumo innecesario de recursos y energía y hace que su actuación sea mucho más eficiente. Además, la detección de amenazas y riesgos a través de este sistema reduce las pérdidas de material y equipamiento que puedan derivarse de un ataque.

Por otro lado, en la parte que concierne a este TFM, los sistemas que se necesitan no implican el uso de nuevas máquinas ya que se pueden usar sistemas de los que toda organización dispone, reduciendo las emisiones que supone la construcción de nuevos edificios que las alberguen.

### **Impacto ético**

En cuanto a los impactos éticos derivados del sistema, se pueden destacar los siguientes:

- Soporte a instituciones.
- Privacidad de la Información.

El uso de las nuevas tecnologías supone un gran impacto sobre la sociedad. El creciente interés en la ciberseguridad engloba tanto a las personas desde el punto de vista individual como a los organismos y empresas. Los organismos institucionales pueden almacenar información confidencial y/o información de los ciudadanos de un país. Si alguno de estos organismos es víctima de un ataque, entonces esa información puede verse comprometida, así como la privacidad de cada una de las personas a las que pertenece dicha información. Por eso, con la implantación de este sistema, se trata de identificar posibles problemas y se monitoriza el sistema de la institución para poder protegerla.

frente incidentes que atenten contra la privacidad. Como se ha mencionado a lo largo del trabajo, compartir información sobre amenazas entre organismos nacionales e internacionales también supone el enriquecimiento de la información que se tiene en la organización sobre las amenazas. En este trabajo se hace uso de estándares para favorecer esta compartición. Esto conlleva poder anticiparse a los incidentes que puedan afectar a la organización y luchar contra las APTs que cada vez son más frecuentes.

Por otro lado, la privacidad de la información también es importante en este trabajo puesto que la información que se procesaría en un caso real pertenecerá a datos procedentes de sensores que monitorizan tanto los sistemas como el comportamiento de personas en esos sistemas. La RGDP regula el tratamiento de los datos personales. Esta regulación se aplica a cualquier institución que trate datos de personas pertenecientes a la Unión Europea con el fin de proteger sus derechos y libertades.

En el caso concreto de la información usada en este trabajo, se ha utilizado información hipotética, por lo que no aplica. Sin embargo, en un entorno real cabe destacar que el sistema se utiliza como un medio para la empresa responsable para cumplir sus obligaciones frente a la protección de datos. El sistema ayudará al responsable a realizar el análisis de riesgos, su identificación y tratamiento. El desarrollo de este trabajo permite detectar amenazas que van contra del Cumplimiento de la Protección de Datos. Por tanto, este sistema ofrecerá a la empresa responsable la capacidad de garantizar la seguridad de sus datos personales y, conforme al art.32, asegurar la confidencialidad, integridad y disponibilidad de los datos.

### A.3 CONCLUSIONES

Para la implementación de un proyecto se debe realizar un análisis de los impactos derivados de su implantación. El trabajo que aquí ocupa presenta impactos relevantes en el ámbito socioeconómico, ético y ambiental.

Con la implantación del sistema desarrollado se responde de manera proactiva a las amenazas, lo que disminuye el riesgo al que están expuestas las organizaciones y genera un impacto positivo económico al reducirse los efectos negativos sobre los sistemas de la organización como consecuencia de un incidente. Además, el sistema reduce el consumo energético y los costes necesarios para paliar las consecuencias que produce la materialización de las amenazas.

Por otro lado, el uso de estándares facilita la compartición de información entre organizaciones, por lo que el sistema se presenta como un soporte para las instituciones, ofreciendo beneficios sobre la detección temprana de amenazas. Además, supone una evolución de los sistemas de las organizaciones hacia sistemas expertos que supone la automatización de sus procesos de ciberseguridad.

Con todo ello, se ha concluido que el proyecto desarrollado tiene un gran impacto tanto económico y social, como ético y ambiental y favorece la innovación y evolución de la ciberseguridad.

## ANEXO B: PRESUPUESTO ECONÓMICO

---

En este apartado se analizan los gastos económicos necesarios para la realización del trabajo. Para ello, se ha realizado un presupuesto económico. Los costes a tener en cuenta son:

- Coste de recursos humanos
- Coste de recursos materiales

El trabajo se ha realizado con una duración estimada de 7 meses que se han dividido de la siguiente manera: 2 meses dedicados al análisis y estudio del estado del arte y el alcance del trabajo, 4 meses para la implementación, desarrollo y pruebas de validación y, por último, un mes para el desarrollo de la documentación teórica y técnica en este documento.

### Recursos humanos

Se diferencia entre recursos humanos para el desarrollo del sistema y recursos humanos para el mantenimiento del sistema.

La persona encargada del desarrollo del proyecto es un Ingeniero Superior en Ingeniería de Tecnologías y Servicios de la Telecomunicación. Esta persona trabaja 750 horas durante 7 meses. El trabajo desempeñado por esta persona consiste en el desarrollo y prueba del sistema.

Para el mantenimiento del sistema se necesita un Ingeniero con conocimiento de Ciberseguridad. El precio medio de un trabajador con estas características por hora es de 15 €/hora.

Por tanto, la estimación del coste en recursos humanos para el sistema es el que se observa en la siguiente tabla.

PUESTO	HORAS	PRECIO/HORA	TOTAL
INGENIERO SUPERIOR	750	15,00 €	11.250,00 €

Tabla 34. Costes de recursos humanos

### Recursos materiales

Se realiza la división de manera análoga al caso anterior. Se diferencia entre recursos físicos necesarios para el desarrollo del sistema y recursos físicos para el mantenimiento del mismo.

Para desarrollar el sistema se ha necesitado un único ordenador personal. El resto del software es libre y gratuito, como se ha mencionado anteriormente y no existen gastos de licencias de software. El ordenador personal presenta las siguientes especificaciones:

- 16 GB de RAM.
- Intel Core i7 2.9 GHz.
- 30GB de espacio en disco.
- Sistema operativo Mac OS.

Si bien el precio de un PC de estas características ronda los 2000 €, se pueden utilizar otros más económicos con característica similares de valor de 1000€. Sin embargo, para calcular el coste total se va a tener en cuenta el material utilizado.

Para el mantenimiento del sistema depende del entorno de la organización en cuestión, ya que en función de una u otra existirán más activos, redes y como consecuencia la operación de cómputo se incrementará.

La siguiente tabla muestra los gastos en el desarrollo del sistema debido a los recursos físicos utilizados. Se ha considerado la amortización estándar de 5 años para los ordenadores personales, así como un periodo de 7 meses.

MATERIAL	PRECIO	USO EN MESES	AMORTIZACIÓN (EN AÑOS)	TOTAL
<b>ORDENADOR PERSONAL(SOFTWARE INCLUIDO)</b>	2.000,00 €	7	5	233,33 €
<b>PROTÉGÉ (LICENCIA LIBRE)</b>	-	7	-	-
<b>ECLIPSE (LICENCIA LIBRE)</b>	-	7	-	-

Tabla 35. Costes de recursos materiales

En este apartado también se tiene en cuenta el material de oficina necesarios:

TIPO	TOTAL
<b>IMPRESIÓN</b>	100,00 €

Tabla 36. Costes de material fungible

### Costes generales y beneficio industrial

La siguiente tabla refleja los costes generales y el beneficio industrial. Para el cálculo de los costes generales se tiene en cuenta un 15 % sobre los costes directos. Se considera un beneficio industrial de un 6 % sobre los costes directos e indirectos.

<b>Gastos Generales</b>	15%	Sobre Costes directos	1.722,50 €
<b>Beneficio Industrial</b>	6%	Sobre CD y CI	792,35 €

Tabla 37. Costes generales y beneficio industrial

### Coste en Licencias

Como se ha mencionado anteriormente, el coste en licencias es nulo porque se utiliza software libre.

**Presupuesto total**

La siguiente tabla refleja el presupuesto del trabajo antes de aplicar impuestos y el gasto en impuestos que debe ser aplicado. Una vez aplicados dichos impuestos se muestra el presupuesto total final que implica la implementación y el desarrollo de todo el sistema.

<b>Presupuesto Subtotal</b>		14.098,18 €
<b>IVA Aplicable</b>	21%	2.960,62 €
<b>Total</b>		<b>17.058,80 €</b>

Tabla 38. Presupuesto total antes y después de impuestos

