

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA  
DE TECNOLOGÍAS Y SERVICIOS DE  
TELECOMUNICACIÓN  
TRABAJO FIN DE GRADO**

**DISEÑO Y DESARROLLO DE UN  
SISTEMA DE VISUALIZACIÓN  
EN TIEMPO REAL DE RIESGO  
DE CIBERSEGURIDAD**

**CARLOS AZNAR OLMOS**

**2020**

# GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

## TRABAJO FIN DE GRADO

**Título:** Diseño y desarrollo de un sistema de visualización en tiempo real de riesgo de ciberseguridad

**Autor:** D. Carlos Aznar Olmos

**Tutor:** D. Víctor A. Villagrà González

**Ponente:** D. ....

**Departamento:** Ingeniería Telemática

## MIEMBROS DEL TRIBUNAL

**Presidente:** D. ....

**Vocal:** D. ....

**Secretario:** D. ....

**Suplente:** D. ....

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

Madrid, a                      de                      de 20...

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA  
DE TECNOLOGÍAS Y SERVICIOS DE  
TELECOMUNICACIÓN  
TRABAJO FIN DE GRADO**

**DISEÑO Y DESARROLLO DE UN  
SISTEMA DE VISUALIZACIÓN  
EN TIEMPO REAL DE RIESGO  
DE CIBERSEGURIDAD**

**CARLOS AZNAR OLMOS  
2020**

Agradecido a mi familia por creer en mí durante todo este tiempo y jamás darse por vencidos.

A mis amigos por apoyarme en los momentos más complicados.

A Nerea y Juan por estar ahí en lo bueno y en lo malo.

A mi tutor, Víctor, por ayudarme en todo lo posible para que esto salga adelante

Simplemente, gracias. ■

## RESUMEN

La tecnología se ha convertido en una parte esencial de nuestras vidas debido a que en cualquier momento estamos conectado a un dispositivo con acceso a Internet, ya sea en el trabajo mediante el uso del ordenador o en nuestro tiempo libre por medio del teléfono móvil. Esta exposición continua a Internet provoca que dichos dispositivos puedan sufrir ataques cibernéticos los cuales pueden substraer información personal. Para evitar este ataque es necesario proteger nuestros equipos ya sean personales o de trabajo. Por otra parte, no solo el robo de información se produce por ataques a través de Internet, sino que también se pueden producir de forma física mediante suplantación de identidad o por el método de ingeniería social. Es decir, por una manera o por otra estamos continuamente en riesgo.

Para poder evaluar y visualizar este riesgo se ha partido de un sistema ya desarrollado mediante el cual, por medio de un razonador semántico, es capaz de analizar unas amenazas entrantes y a través de una serie de reglas SWRL y empleando la tecnología OWL, calcula un nivel de riesgo. Este nivel de riesgo se calcula en tiempo real para todas las Comunidades Autónomas que hay en España incluyendo las Islas Baleares, las Islas Canarias y las Ciudades Autónomas de Ceuta y Melilla. A parte de calcular el riesgo a nivel autonómico, también permite evaluarlo para una serie de activos correspondientes a cada comunidad.

Se realizará un sistema de visualización en tiempo real que permitirá la representación de los distintos niveles de riesgo sobre un mapa de España. Este riesgo se clasifica en tres niveles dependiendo del grado de amenaza: nivel bajo representado por medio del color verde, nivel medio representado por el color amarillo y nivel alto representado por el color rojo. A su vez en cada nivel habrá distintos subniveles para un mejor análisis. Esto permitirá a los usuarios conocer en tiempo real, si sus equipos se encuentran o no en riesgo de ser víctimas de un ataque.

El proyecto finalizará con una propuesta de mejoras para líneas futuras las cuales consistirán en realizar análisis de niveles de riesgo también a nivel provincial y mejorar la visualización de estos haciendo que ésta sea más interactiva.

Finalmente, cabe destacar que el programa desarrollado, así como su visualización, no garantiza la seguridad en los dispositivos propensos a sufrir un ataque cibernético ya que lo que mide la herramienta es la probabilidad con la que se puede producir dicho ataque. Por lo tanto, para reducir los daños producidos por un ataque, es necesario proteger bien todos los dispositivos electrónicos, así como de disponer de seguros frente a la extracción de datos privados o personales ya que esto puede tener consecuencias futuras como, por ejemplo, obtener una mala imagen o reputación en el caso de que una empresa sufra la expropiación de los datos de sus clientes.

## PALABRAS CLAVE

OWL, SWRL, tiempo real, riesgo, amenazas, ataque, visualización, Comunidades Autónomas, ontología, datos, activos, empresas, razonamiento.

## SUMMARY

Technology has become an essential part of our lives because at any time we are connected to a device with Internet access, whether at work using the computer or in our free time through the mobile phone. This continuous exposure to the Internet makes such devices susceptible to cyber-attacks which can steal personal information. To prevent these attacks, we need to protect our computers, whether they are personal or work related. Moreover, not only does information theft occur through Internet attacks, but it can also occur physically through phishing or social engineering. In other words, in one way or another we are continually at risk.

In order to evaluate and visualize this risk, we have started with a system already developed by which, by means of a semantic reasoner, it is capable of analyzing some incoming threats and through a series of SWRL rules and using OWL technology, it calculates a risk level. This risk level is calculated in real time for all the Autonomous Communities in Spain, including the Balearic Islands, the Canary Islands and the Autonomous Cities of Ceuta and Melilla. In addition to calculating the risk at an autonomous community level, it also allows it to be evaluated for a series of assets corresponding to each community.

A real-time visualization system will be implemented that will allow the representation of the different risk levels on a map of Spain. This risk is classified into three levels depending on the degree of threat: low level represented by the colour green, medium level represented by the colour yellow and high level represented by the colour red. At each level there will be different sub-levels for a better analysis. This will allow users to know in real time, if their computers are at risk of being attacked or not.

The project will end with a proposal of improvements for future lines which will consist of making analysis of risk levels also at a provincial level and improving the visualization of the same making it more interactive.

Finally, it should be noted that the program developed, as well as its visualization, does not guarantee security on devices prone to cyber-attacks, since what the tool measures is the probability with which such an attack can occur. Therefore, in order to reduce the damage produced by an attack, it is necessary to protect all electronic devices as well as to have insurance against the extraction of private or personal data since this can have future consequences such as obtaining a bad image or reputation in the event that a company suffers the expropriation of its clients' data.

## KEYWORDS

OWL, SWRL, real time, risk, threats, attack, visualization, Autonomous Communities, ontology, data, assets, companies, reasoning.

# ÍNDICE DEL CONTENIDO

<b>1. INTRODUCCIÓN Y OBJETIVOS .....</b>	<b>1</b>
1.1. INTRODUCCIÓN.....	1
1.2. OBJETIVOS .....	1
1.3. ESTRUCTURA DE LA MEMORIA.....	2
<b>2. ESTADO DEL ARTE .....</b>	<b>3</b>
2.1. CIBERSEGURIDAD .....	3
2.2. LENGUAJE DE ONTOLOGÍAS WEB (OWL) .....	4
2.2.1. MÉTRICAS DE RIESGOS (SWRL) .....	5
2.2.2. PROPIEDADES OWL.....	6
2.3. MÉTRICAS DE RIESGOS (SWRL) .....	6
<b>3. ARQUITECTURA .....</b>	<b>8</b>
3.1. ETAPAS DEL DISEÑO .....	8
3.2. REQUISITOS FUNCIONALES .....	10
<b>4. DESARROLLO .....</b>	<b>12</b>
4.1. RAZONAMIENTO DE LAS AMENAZAS.....	12
4.1.1. CONFIGURACIÓN INICIAL.....	12
4.1.2. ONTOLOGÍAS Y REGLAS SWRL .....	13
4.1.3. RAZONADOR .....	13
4.2. OBTENCIÓN DE DATOS .....	14
4.2.1. CONFIGURACIÓN DEL SISTEMA .....	15
4.2.2. GUARDADO Y ACTUALIZACIÓN DE DATOS .....	18
4.3. VISUALIZACIÓN .....	22
4.3.1. LECTURA DE DATOS .....	22
4.3.2. REPRESENTACIÓN DE LOS DATOS.....	23
NIVELES DE RIESGO .....	23
<b>5. RESULTADOS.....</b>	<b>26</b>
5.1. RESULTADOS DE LOS ACTIVOS .....	26
5.2. RESULTADOS DE LOS DATOS GUARDADOS.....	28
5.3. RESULTADOS DE LA VISUALIZACIÓN.....	29
5.3.1. RESULTADOS DE RIESGO BAJO .....	29
5.3.2. RESULTADOS DE RIESGO MEDIO .....	30
5.3.3. RESULTADOS DE RIESGO ALTO .....	32

<b>6. CONCLUSIONES Y LÍNEAS FUTURAS.....</b>	<b>34</b>
6.1. CONCLUSIONES .....	34
6.2. LÍNEAS FUTURAS .....	35
<b>7. BIBLIOGRAFÍA .....</b>	<b>36</b>
<b>ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES .....</b>	<b>38</b>
A.1 INTRODUCCIÓN .....	38
A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO .....	38
A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS.....	39
A.4 CONCLUSIONES.....	39
<b>ANEXO B: PRESUPUESTO ECONÓMICO .....</b>	<b>2</b>



## ÍNDICE DE FIGURAS

Figura 1: Representación de Individuos .....	5
Figura 2: Representación de propiedades .....	5
Figura 3: Representación de una Clase que contiene Individuos.....	6
Figura 4: Ejemplo regla SWRL .....	7
Figura 5: Arquitectura del Sistema .....	8
Figura 6: Programa de obtención de datos de la ontología .....	9
Figura 7: Resultados de datos obtenidos de la ontología .....	9
Figura 8: Resultado fichero Datos_Andalucia.json .....	11
Figura 9: Resultado del análisis de la ontología.....	14
Figura 10: Resultado de un activo de Aragón.....	15
Figura 11: Archivo Configuracion.txt.....	15
Figura 12: Ruta del archivo de Configuración.txt.....	16
Figura 13: Clase ConfigurationFilePath.java .....	17
Figura 14: Clase Configuration.java .....	17
Figura 15: Creación del modelo ontológico. Clase UpdateRealTimeData.java .....	18
Figura 16: Recorrido de la ontología: Clase UpdateRealTimeData.java .....	19
Figura 17: Guardado de fichero. Clase UpdateRealTimeData.java .....	20
Figura 18: Actualización de datos. Clase UpdateRealTimeData.java .....	21
Figura 19: Inicio Clase UpdateRealTimeData.java .....	21
Figura 20: Guardado fecha y hora de la última actualización.....	22
Figura 21: Librería JQuery y JQuery Maphilight .....	22
Figura 22: Función de obtención de datos de Andalucía .....	23
Figura 23: Niveles de riesgo .....	24
Figura 24: Obtención datos última actualización.....	25
Figura 25: Resultado recorrido ontología. Obtención datos del riesgo por CCAA .....	26
Figura 26: Resultado ontológico del riesgo de Ceuta .....	27
Figura 27: Resultado ontológico del riesgo de Extremadura .....	27
Figura 28: Resultado activos de Castilla y León.....	27
Figura 29: Resultados de la ejecución del programa y del guardado de datos.....	28
Figura 30: Resultado mapa de España con riesgo bajo.....	29
Figura 31: Resultado activos de Castilla y León con riesgo bajo .....	30
Figura 32: Resultado mapa de España con riesgo medio.....	30
Figura 33: Resultado activos de Castilla y León con riesgo medio .....	31
Figura 34: Resultado mapa de España con riesgo alto.....	32
Figura 35: Resultado activos de Castilla y León con riesgo alto .....	33

# 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1. INTRODUCCIÓN

Hoy en día vivimos en una sociedad en la que prácticamente todo está conectado a través de Internet. Por consiguiente, nuestras empresas son las más vulnerables a recibir ciberataques con el objetivo de obtener información y datos acerca de la empresa.

En España el 98% de las empresas utilizan Internet en su día a día y dentro de ese porcentaje el 76% del tiempo es empleado en el uso de herramientas tecnológicas (1). Esto unido a que los empleados y las empresas tienden a almacenar toda la información, ya sea en local o en la nube, la preocupación por un ataque cibernético es clave para las organizaciones.

No todas las amenazas que puede sufrir un equipo tienen el mismo nivel de gravedad ni afectan de la misma manera a las infraestructuras atacadas. Estas amenazas pueden clasificarse en amenazas internas las cuales son aquellas causadas desde la propia red ya sea por usuarios o por personal técnico y amenazas externas son aquellas que se originan fuera de la red local ya sea debido a vulnerabilidades en la propia red que permiten tener acceso a la misma o mediante ataques realizados por personas ajenas (2).

Como hemos dicho antes, ya que actualmente estamos continuamente expuestos a ser víctimas de un ciberataque el cual puede producir no solo daños en los equipos afectados sino también puede suponer un problema de reputación o de imagen para la empresa que lo sufre, es necesario conocer los riesgos a los que estamos expuestos continuamente, medirlos y evaluarlos para reducir en todo lo posible los daños que se puedan producir. Sin embargo, no siempre se va a poder controlar todos los riesgos por lo que lo más eficaz y seguro para una empresa u organización, aparte de disponer en sus equipos de sistemas que te alerten de que puedes sufrir ataques, es necesario que dispongan de seguros de ciberriesgos cuyo objetivo, como hemos mencionado anteriormente, es mitigar en todo lo posible los efectos producidos al ser víctimas de un ataque malicioso.

Una solución muy práctica que se puede implementar para poder evaluar estos riesgos es realizar una representación visual en la cual se podría observar el grado de amenaza al que nuestros equipos se encuentran expuestos. Además, si esta visualización se puede realizar de dos maneras: en *near-real time* en la cual el usuario conocerá, prácticamente en tiempo real, los resultados obtenidos y así estar constantemente controlando el riesgo al que está expuesta una comunidad, una empresa o un individuo. La segunda forma de visualización es cada vez que el usuario desee. Con este tipo de visualización el usuario tendrá el control de, cuando él lo desee, actualizar los datos mediante un archivo de configuración el cual el usuario puede poner el tiempo de actualización de los datos.

## 1.2. OBJETIVOS

El objetivo principal es desarrollar un programa que permita conocer en tiempo real la visualización de los posibles riesgos que tienen las empresas en las diferentes comunidades autónomas dentro del territorio nacional. Esta visualización se implantará sobre un mapa interactivo en el cual se podrá ver el riesgo que tiene cada comunidad además de obtener información acerca de las diferentes empresas que hay en la comunidad divididas por activos, es decir, por ciudadanos, por empresas o por investigación y universidad.

Para poder conseguir este objetivo primero es necesario obtener el riesgo de las diferentes comunidades autónomas a partir de ontologías. Este riesgo está previamente calculado a través del uso de razonadores semánticos mediante multiprocesamiento. Cabe destacar que este razonamiento sobre amenazas se hace en *near-real time* por lo tanto la visualización que debemos implementar también tiene que ser del mismo estilo. Una vez que obtenemos los resultados sobre el análisis de las amenazas, lo tenemos que representar de manera visual sobre un mapa de España,

más concretamente, al tratarse de riesgos específicos por comunidad autónoma, la visualización se realizará sobre las diferentes comunidades que hay en España incluyendo las Islas Baleares, las Islas Canarias y las Ciudades Autónomas de Ceuta y Melilla.

Por otra parte, existen algunos objetivos secundarios necesarios para que se cumpla el objetivo principal. El primero de ellos es la obtención del riesgo que existe en cada comunidad, así como el resultado de los diferentes individuos de una determinada comunidad autónoma. Dichos datos serán guardados en ficheros JSON para su posterior visualización debido a que es más fácil trabajar con este tipo de ficheros.

El segundo objetivo secundario es el anterior a la visualización y consiste en la lectura correcta de los ficheros JSON, que contienen tanto el riesgo de la comunidad como el riesgo de los distintos individuos que forman parte de la comunidad. Cabe destacar que existirán tantos ficheros como número de comunidades autónomas haya debido a que así la lectura de los resultados se realiza de manera más sencilla su representación será mucho más ordenada y visual.

Una vez cumplido los objetivos secundarios hay que cumplir el objetivo principal que es la visualización en tiempo real del riesgo en las diferentes comunidades autónomas. Este objetivo principal consiste en una cohesión de los dos objetivos secundarios anteriormente descritos. Para poder actualizar los datos en tiempo real, el programa realizado para la obtención de estos tiene que estar constantemente en ejecución y cuyo funcionamiento se explicará más adelante en el apartado 4.2 (Obtención de datos). Al estar este programa constantemente funcionando, los datos que guarda están constantemente actualizándose por lo tanto la representación que se haga sobre el mapa de España se realizará en *near-real time* cumpliéndose así con el objetivo principal del trabajo. Sin embargo, en la práctica guardar los datos en tiempo real es prácticamente imposible debido al tiempo que tarda en programa en actualizar los datos, por lo tanto, para realizar un programa que se asemeje lo máximo al objetivo, se creará un archivo de configuración mediante el cual el usuario podrá elegir el intervalo de tiempo que se actualicen los resultados obtenidos.

### 1.3. ESTRUCTURA DE LA MEMORIA

Para explicar el desarrollo del proyecto y así poder conseguir con los objetivos fijados en el apartado anterior, la memoria se realizará siguiendo la siguiente estructura:

1. Introducción: Se realiza una breve explicación sobre en qué consiste el proyecto, así como los objetivos a cumplir.
2. Estado del arte: Se describe las tecnologías o aplicaciones tanto actuales como pasadas con las que se desarrollará el trabajo.
3. Arquitectura: Se indica y se explica la arquitectura software escogida para la realización del proyecto.
4. Desarrollo: Se expone toda arquitectura y desarrollo llevado a cabo en la realización del proyecto.
5. Resultados: Se analiza los resultados obtenidos durante la parte de desarrollo y se llega a una conclusión acerca del trabajo realizado.

## 2. ESTADO DEL ARTE

El riesgo que tienen las empresas a sufrir ciberataques se debe a que todas ellas disponen de dispositivos electrónicos conectados a Internet, ya sea los ordenadores de las propias empresas o los dispositivos móviles de los trabajadores u otros sistemas que dispongan de conexión a Internet ya sea impresoras, faxes o routers. Para poder controlar el riesgo y reducir el impacto de estos ataques al mínimo posible debemos tener en cuenta los siguientes aspectos:

### 2.1. CIBERSEGURIDAD

Actualmente la ciberseguridad desempeña un papel fundamental en nuestra sociedad debido a que vivimos en una época en la que prácticamente todo está informatizado y funciona a través de dispositivos electrónicos, es decir, podemos realizar cualquier cosa que se nos ocurra solo utilizando nuestro dispositivo móvil ya sea realizar una compra, pagar un recibo, realizar una transferencia bancaria... Por consiguiente, estamos más expuestos a recibir ciberataques y que nos roben toda la información que almacenamos en nuestros dispositivos.

Por este motivo los ciberataques se han convertido en una de las principales amenazas para las empresas por la gran cantidad de datos que manejan. En consecuencia, para prevenir estos ataques y minimizar sus consecuencias al mínimo las empresas invierten cada vez más en ciberseguridad.

Uno de los ataques más habituales es el denominado *ransomware* o secuestro de datos. El ataque consiste en la encriptación de la información por un ciber-delincuente que impide el acceso y solicita una cantidad económica para su desencriptación. La realización de este ataque se realiza por medio de un software malicioso que infecta el ordenador o dispositivo de la sede donde se ejecuta. Por este motivo, las empresas han empezado a establecer mecanismos de análisis y gestión de riesgos para evitar ataques a la seguridad de la información.

La gestión de riesgos de ciberseguridad se debe entender como uno de los pilares fundamentales para salvaguardar la confidencialidad e integridad de los activos de información, infraestructuras críticas y datos personales. Para que una empresa consiga gestionar con éxito cada riesgo, deberá acudir a algunas normas como el ISO 31000:2018 o el ISO/IEC 27005:2001 (3). También está la metodología de análisis y gestión de riesgos de los sistemas de información (MAGERIT), la cual consiste en realizar el análisis de riesgos como una aproximación metódica para determinar el riesgo siguiendo unos pasos (4):

1. Determinación de los activos relevantes para la organización, su interrelación y su valor, en el sentido de qué perjuicio supondrá su degradación.
2. Determinar a qué amenazas están expuestas esos activos.
3. Determinar qué salvaguardas hay dispuestas y como de eficaces resultan frente al riesgo.
4. Estimar el impacto mediante el daño producido sobre el activo derivado de la materialización de la amenaza.
5. Estimar el riesgo, definido como el impacto ponderado con la tasa de ocurrencia de la amenaza.

En relación con los impactos y riesgos a los que están expuesto el sistema, es necesario tomar una serie de decisiones condicionadas por diversos factores para la gestión de estos riesgos (4):

- La gravedad de impacto o del riesgo
- La obligación a la que por ley está sometida la organización
- Las obligaciones a la que los reglamentos sectoriales esté sometida la organización
- Las obligaciones a los que por contrato esté sometida la organización

Todas las consideraciones anteriores desembocan en una calificación de cada riesgo determinando si es (4):

1. **Crítico:** Se requiere atención urgente.
2. **Grave:** Se requiere atención.
3. **Apreciable:** Puede ser objeto de estudio para su tratamiento
4. **Asumible:** No se van a tomar acciones para atajarlo.

El análisis de riesgos determina impactos y riesgos. Los impactos recogen daños absolutos, independientemente de si es o no más probable de que se produzca la circunstancia, esto es, refleja el daño posible. Sin embargo, los riesgos ponderan la probabilidad de que ocurra, es decir, reflejan el daño probable (4).

Por todos estos motivos se creó un proyecto denominado proyecto MODRIC en colaboración con el Instituto Nacional de Ciberseguridad (INCIBE) (5) (6), sobre el cual se basa mi trabajo, que consiste en un prototipo diseñado para modelizar el riesgo de los dominios administrativos, ejemplificando el caso de un país en tiempo real, más concretamente España, es decir, permitirá la medición y el cálculo dinámico de las métricas de riesgo de ciberseguridad en un dominio administrativo con sólo algunas estimaciones de los activos. Para ello se ha realizado un modelado de dichos activos y amenazas detectadas por diversas fuentes de información. A su vez, toda esta información se almacena en forma de conocimiento haciendo uso de ontologías lo que permite aplicar motores de razonamiento para inferir nuevos conocimientos que puedan ser utilizados posteriormente en el siguiente razonamiento.

Por otra parte, el análisis y gestión de riesgos es una herramienta muy importante en el entorno de la ciberseguridad para supervisar los diferentes atributos de un entorno interactivo con el fin de determinar las posibles vulnerabilidades a las que está sometido el entorno y, por tanto, el estado de seguridad del medio ambiente. Hoy en día la gestión de riesgos busca evolucionar los enfoques clásicos del análisis estático de riesgos. Para ello, estos enfoques suelen tomar una instantánea de la de la situación de organización en un determinado momento y generar métricas y políticas de seguridad (5) (6). Sin embargo, la mayoría de los parámetros que intervienen en un análisis de riesgos son dinámicos, es decir, cambian constantemente a lo largo del tiempo. Estos sistemas dinámicos intentan adaptarse a estas circunstancias incluyendo variaciones temporales en los elementos que componen este análisis. No obstante, este sistema dinámico puede resultar complejo para el cálculo de riesgo en un entorno global como un país, una región o un sector público debido a que los activos del entorno no están claramente definidos como pertenecientes a un dominio administrativo. Por este motivo se desarrolló el modelo formal (5) (6) anteriormente descrito basado en un sistema ontológico en el que se define todos los activos involucrados para definir métrica de seguridad adecuadas que calculan los diferentes niveles de riesgo para el dominio o subdominios.

Como conclusión realizar un análisis de riesgos es importante porque nos permite identificar los principales riesgos existentes en nuestra organización. Una vez que hayamos realizado este análisis e identificado las posibles amenazas, el siguiente paso es centrarse en la gestión de esos riesgos y para ello es necesario establecer un umbral a partir del cual indicaremos que riesgos son asumibles y cuáles no. Finalmente es necesario definir un plan de tratamiento de estos riesgos que recoja las acciones que se llevarán a cabo para controlarlos.

## 2.2. LENGUAJE DE ONTOLOGÍAS WEB (OWL)

Empezaremos por la definición de ontología, la cual se utiliza para capturar el conocimiento sobre algún dominio de interés. Una ontología es una definición formal de tipos, propiedades y relaciones entre entidades que existen para un dominio en particular. En los campos de la Ingeniería de Sistemas, Ingeniería de Software, Ingeniería Biomédica, Ingeniería Artificial y Arquitectura de la Información se crean ontologías para limitar la complejidad y para organizar la información, por lo que puede ser aplicada a la resolución de problemas.

Existen varios tipos de ontología, sin embargo, nos vamos a centrar en la ontología de información la cual especifica la estructura de almacenamiento de base de datos.

El lenguaje que se usa para leer las ontologías es OWL de World Wide Web Consortium (W3C). Este lenguaje sirve para hacer declaraciones ontológicas como seguimiento de RDF y RDFs (7). Al igual que la herramienta *Protégé*, OWL hace posible describir conceptos, pero también proporciona nuevas facultades. Por otra parte, el lenguaje OWL está basado en un modelo lógico el cual permite que conceptos complejos puedan ser contruidos a partir de conceptos más simples.

### 2.2.1. MÉTRICAS DE RIESGOS (SWRL)

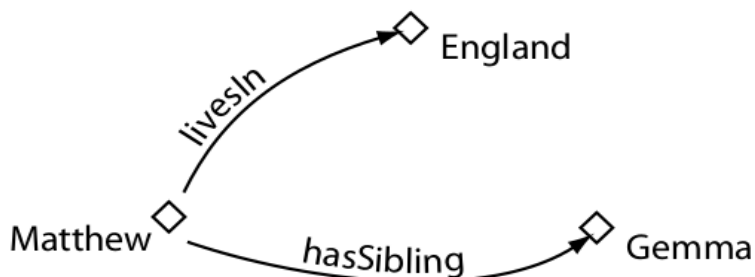
La ontología OWL presenta, como hemos dicho anteriormente, componentes similares a las ontologías basadas en *Protégé*, sin embargo, existe una pequeña diferencia y es en la manera que se utiliza pasara describir estos componentes. Mientras que *Protégé* emplea Instancias (Instances), Slots (Slots) y Clases, (Classes), OWL consiste en Individuos (individuals), Propiedades (Properties) y Clases (Classes) (8):

- Individuos o *Individuals*: Representan objetos en el dominio en el que estamos interesados. En OWL se debe establecer que los individuos son iguales o diferentes entre sí ya que esto podría llevar a errores. Por otro lado, los Individuos pueden referirse como “Instancias de una clase” (8) .



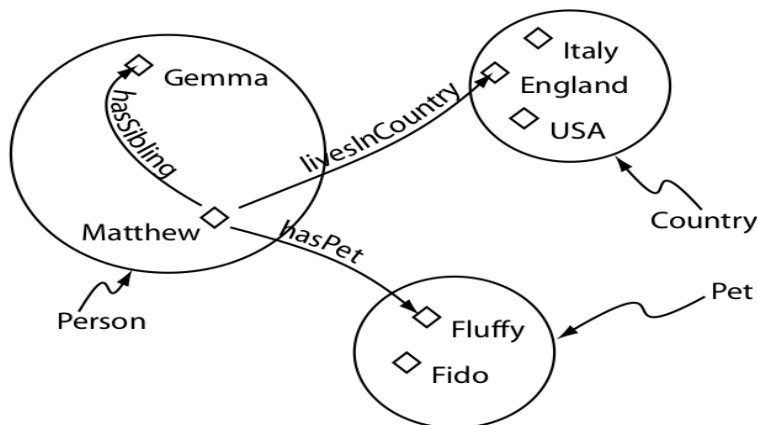
**Figura 1: Respresentación de Individuos**

- Propiedades o *Properties*: Son relaciones binarias de individuos, es decir, son relaciones entre dos individuos. Las propiedades pueden tener inverso y estas a su vez pueden ser transitivas o simétricas (8).



**Figura 2: Representación de propiedades**

- **Calses o Clases:** Son conjuntos que contienen individuos. Estas clases pueden estar organizadas por jerarquías de Superclases y Subclases. Una Subclase está incluida dentro de una Superclase. En OWL las clases están compuestas por descripciones que especifican las condiciones que debe cumplir un individuo para que se miembro de la clase (8).



**Figura 3: Representación de una Clase que contiene Individuos**

#### 2.2.2. PROPIEDADES OWL

Las propiedades OWL representan relaciones. Existen dos tipos de propiedades (7):

- **Propiedades de objetos u *Object properties*:** Son las relaciones entre dos individuos, esto es, une un individuo con otro individuo.
- **Propiedades datos o *Datatype properties*:** Describe la relación entre un individuo y un valor de datos asociados.

A veces OWL puede presentar una tercera propiedad (*Annotation propertie*) la cual se usa para añadir información a las clases, a los individuos y a la propiedad de datos.

### 2.3.MÉTRICAS DE RIESGOS (SWRL)

El lenguaje de Reglas de la Web Semántica cuyas siglas en ingles son SWRL (Semantic, Web Rule Language) es un lenguaje de reglas basado en OWL y de RuleML (RuleMarkup Language) (9). SWRL permite a los usuarios escribir reglas que pueden ser expresadas en conceptos OWL para proporcionar capacidades de razonamiento deductivo. Semánticamente SWRL está construido sobre la misma base lógica de descripción que OWL y que proporciona garantías formales fuertes similares cuando se realiza la inferencia. SWRL extiende el conjunto de axiomas de las ontologías OWL para incluir cláusulas Horn pudiendo así combinar reglas de tipo Horn con bases de conocimiento OWL.

Una regla SWRL contiene la forma de implicación entre un antecedente (cuerpo o *body*) y un consecuente (cabeza o *head*). Ambos en conjunto tanto el cuerpo como la cabeza consiste en un conjunto denominado átomos (*atoms*). De forma informal, una regla SWRL se puede interpretar como una indicación de que, si el antecedente es verdadero, entonces el consecuente también lo tiene que ser.



Tanto el antecedente como el consecuente consisten en cero o más átomos los cuales no pueden ser ni negativos ni disjuntos. Un antecedente vacío se trata trivialmente como verdadero, es decir, se satisface para cada interpretación, lo que provoca que el consecuente también debe satisfacerse en cada interpretación. Por el contrario, un consecuente vacío se trata trivialmente como falso lo que implica que ni el antecedente ni el consecuente se satisfacen en interpretación alguna. Por otra parte, los átomos múltiples son tratados como una conjunción.

En SWRL los símbolos predicados puede incluir clases OWL, propiedades o tipos de datos. A su vez, los argumentos pueden ser individuos, valores de datos o variables que los referencia. Todas las variables en SWRL se tratan como universalmente cuantificadas con su ámbito limitando una regla dada.

Existen siete tipos de átomos en SWRL (10):

1. **Clases:** Un átomo clase consiste en una clase OWL nombrada o una expresión de clase y un único argumento representando un individuo OWL.
2. **Propiedades sobre individuos:** Un átomo de propiedad sobre individuos consiste en una propiedad objeto OWL y dos argumentos representando individuos OWL.
3. **Propiedades de valores de datos:** Consiste en una propiedad de datos y dos argumentos, el primero es un individuo y el segundo un valor.
4. **Mismo individuo:** Mediante el símbolo *sameAs* se afirma que los dos argumentos individuos son el mismo.
5. **Individuos diferentes:** Mediante el símbolo *differentFrom* se afirma que los dos argumentos individuos son diferentes.
6. **Rango de datos:** Consiste en un nombre de datos o en un conjunto de literales y un único argumento representando un valor.
7. **Incorporados:** Una de las características que presenta SWRL es la habilidad de soportar incorporaciones construidas por el usuario. Una incorporación es un predicado que toma uno o más argumentos y los envía así si los argumentos satisfacen el predicado.

A continuación, se puede ver una representación de cómo es una regla SWRL empleada para el razonamiento semántico de amenazas en el proyecto. Es representación se corresponde con una amenaza la cual ha de haberse materializado en Aragón y atacar a sistemas basados en Windows Server (5) (6):

```
Modric2:Amenaza(?amenazaAAnalizar) ^ Modric2:Aragón(?amenazaAAnalizar) ^
Modric2:Impacto(?amenazaAAnalizar, ?impact) ^
Modric2:Infraestructura_Crítica(?tipoIndividuo) ^
Modric2:amenaza(?amenazaAAnalizar, Windows_Server) ^
Modric2:ConocimientoCiberseguridad(?tipoIndividuo, ?conocimiento) ^
Modric2:ConfianzaAmenaza(?amenazaAAnalizar, ?confianza) ^
Modric2:PesoServidor(?tipoIndividuo, ?pesoDeDisp) ^
Modric2:Servidor_Windows_Server(?tipoIndividuo, ?porcentajeS0disp) ^
swrlb:multiply(?resultado1, ?impact, ?conocimiento) ^
swrlb:multiply(?resultado2, ?resultado1, ?porcentajeS0disp) ^
swrlb:multiply(?resultado3, ?resultado2, ?confianza) ^
swrlb:multiply(?riesgofinal, ?resultado3, ?pesoDeDisp) ->
Modric2:Riesgo_Servidor(Resultado_Aragón_Empresa_Infraestructura_Crítica, ?riesgofinal)
```

**Figura 4: Ejemplo regla SWRL**



### 3. ARQUITECTURA

La arquitectura del sistema está basada en el uso de razonadores semánticos mediante multiprocesamiento para acelerar dicho razonamiento con el objetivo de conseguir un razonamiento sobre amenazas en *near-real time* y mostrar los resultados de forma gráfica. El esquema del sistema a implementar se muestra en la siguiente imagen:

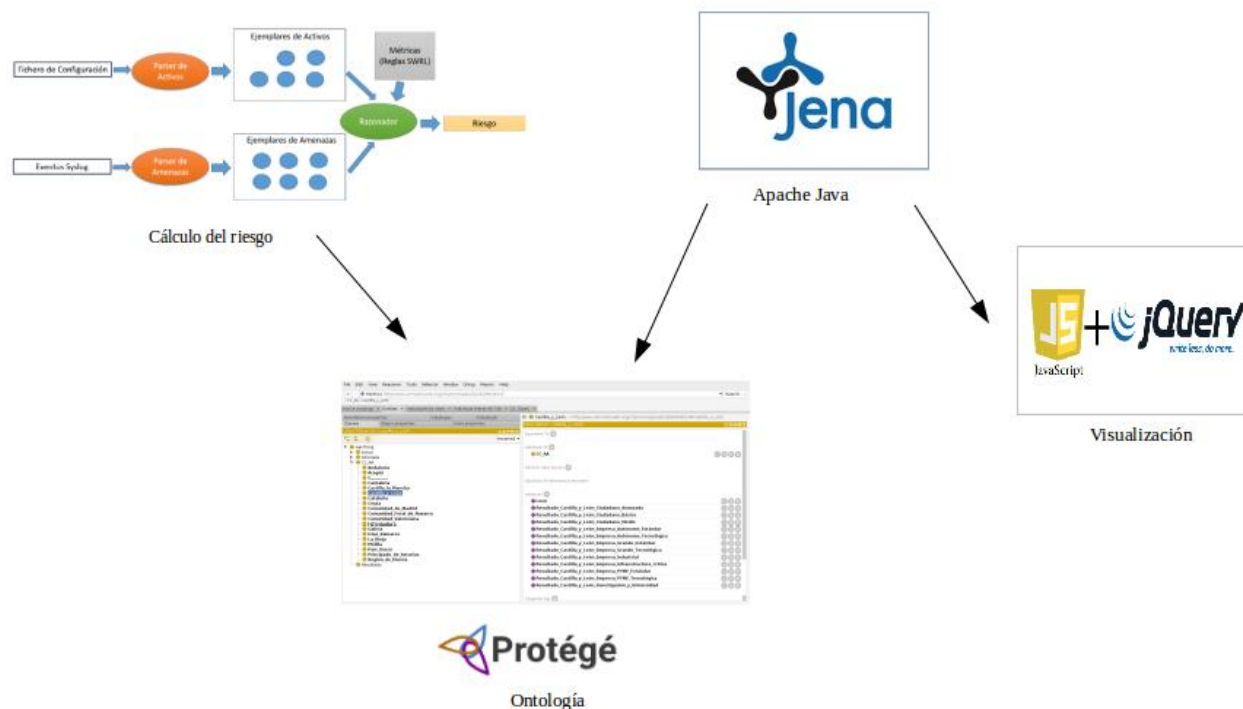


Figura 5: Arquitectura del Sistema

En esta figura se representa la estructura y los procesos a seguir en el desarrollo del proyecto, comenzando por el razonamiento de amenazas empleado y finalizando por la representación gráfica.

#### 3.1. ETAPAS DEL DISEÑO

El sistema completo se puede dividir en tres etapas fundamentales: razonamiento sobre amenazas, obtención de resultados de la ontología y visualización gráfica.

- **Razonamiento sobre las amenazas:** Constituye la primera parte del sistema y la más importante ya que mediante el uso de ontologías y razonadores semánticos se realiza el cálculo de riesgo. En primer lugar, se ha desarrollado una arquitectura a alto nivel en la cual el sistema recibe información de las amenazas a nivel específico y mediante el modelado de unos activos a nivel específico se estima el nivel de riesgo para cada comunidad autónoma.

Para realizar el cálculo de este riesgo primero hay que hacer una configuración inicial en la cual se obtiene los atributos específicos de los activos. A continuación, se lleva a cabo la introducción de la información generada anteriormente en las ontologías utilizando el lenguaje OWL2. Finalmente se introducen reglas SWRL para el cálculo numérico del riesgo por un razonador semántico. Cabe destacar que cada vez que se realiza un cálculo sobre amenazas se crea una copia de la ontología configurada sobre la cual trabajaremos.

- **Obtención de resultados:** Constituye la segunda parte del proyecto y cuyo objetivo es obtener, de la ontología generada anteriormente, el riesgo calculado para cada comunidad autónoma para su posterior representación de forma gráfica.
- **Visualización gráfica:** Antes de poder realizar la visualización es necesario extraer los datos de la ontología. Para realizar dicha extracción se ha empleado la tecnología Java, en concreto Apache Jena especializada en crear aplicaciones basadas en ontologías creando un programa muy sencillo cuyo resultado obtenido es el siguiente:

```
public class ObtenerDatos {
    @SuppressWarnings({ "rawtypes", "unchecked", "unused" })
    public static void main(String [] args) throws IOException {
        //Creamos nuestro modelo
        OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
        model.read("file:/home/carlos/Escritorio/INCIBE-MODRIC/INCIBE/Archivos/Ontologias/Modric2Amenazas.owl", "RDF/XML");

        String amenazasURI = "http://www.semanticweb.org/cris/ontologies/2018/9/Modric2";
        Property anObjectProperty = model.getObjectProperty(amenazasURI);

        String uri = "http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#RiesgoIntermedio";
        Property p = model.getProperty(uri);

        //Recorremos la ontología
        for (Iterator<OntClass> i = model.listHierarchyRootClasses(); i.hasNext();){
            OntClass cls = i.next();
            System.out.print(cls.getLocalName()+" : ");
            for (Iterator it = cls.listInstances(true); it.hasNext();){
                Individual ind = (Individual)it.next();

                if (ind.isIndividual()) {
                    StmtIterator iterStatement = ind.listProperties(p);
                    Statement aStatement = ind.getProperty(anObjectProperty);
                    if (aStatement.getObject().isLiteral()) {
                        while (iterStatement.hasNext()) {
                            System.out.print(ind.getLocalName() + " " + iterStatement.nextStatement().getObject() + "\n");
                        }
                    }
                }
            }
            System.out.println();
        }
    }
}
```

Figura 6: Programa de obtención de datos de la ontología

```
[SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#noProviders for further details.
CC AA: Castilla la Mancha 2.2214579200001254^http://www.w3.org/2001/XMLSchema#double
Andalucía 1.953027193333412^http://www.w3.org/2001/XMLSchema#double
La Rioja 1.6180879866667257^http://www.w3.org/2001/XMLSchema#double
Canarias 1.8194475600000548^http://www.w3.org/2001/XMLSchema#double
Melilla 1.2985720000000227^http://www.w3.org/2001/XMLSchema#double
Comunidad de Madrid 4.8953502000000392^http://www.w3.org/2001/XMLSchema#double
Galicia 1.6451834000001146^http://www.w3.org/2001/XMLSchema#double
Comunidad Valenciana 1.4560059800001106^http://www.w3.org/2001/XMLSchema#double
Principado de Asturias 1.7834376533333582^http://www.w3.org/2001/XMLSchema#double
Castilla y León 3.0622696533334164^http://www.w3.org/2001/XMLSchema#double
Comunidad Foral de Navarra 1.606687250000135^http://www.w3.org/2001/XMLSchema#double
Aragón 3.4292509333336465^http://www.w3.org/2001/XMLSchema#double
Región de Murcia 1.714138300000035^http://www.w3.org/2001/XMLSchema#double
Islas Baleares 2.0270552800000647^http://www.w3.org/2001/XMLSchema#double
Cataluña 1.201664776666742^http://www.w3.org/2001/XMLSchema#double
País Vasco 1.68685626666673^http://www.w3.org/2001/XMLSchema#double
Extremadura 4.298449500000128^http://www.w3.org/2001/XMLSchema#double

Amenaza:
Resultado: Resultado Andalucía Empresa Autónomo Tecnológico 2.0640520000000087^http://www.w3.org/2001/XMLSchema#double
Resultado País Vasco Empresa Grande Tecnológica 1.261866666666714^http://www.w3.org/2001/XMLSchema#double
Resultado Castilla y León Empresa Autónomo Estándar 3.2256000000000087^http://www.w3.org/2001/XMLSchema#double
Resultado Andalucía Empresa PYME Tecnológica 2.1458050000000085^http://www.w3.org/2001/XMLSchema#double
Resultado Castilla la Mancha Empresa Autónomo Estándar 2.1542400000000214^http://www.w3.org/2001/XMLSchema#double
Resultado Ceuta Ciudadano Avanzado 1.4276600000001134^http://www.w3.org/2001/XMLSchema#double
Resultado Andalucía Empresa Grande Tecnológica 1.744746666666741^http://www.w3.org/2001/XMLSchema#double
Resultado Comunidad de Madrid Empresa Industrial 4.6359000000000371^http://www.w3.org/2001/XMLSchema#double
Resultado Comunidad Valenciana Investigación y Universidad 1.10610500000000838^http://www.w3.org/2001/XMLSchema#double
Resultado Principado de Asturias Ciudadano Básico 2.986666666666708^http://www.w3.org/2001/XMLSchema#double
Resultado Aragón Empresa Infraestructura Crítica 2.1060000000001926^http://www.w3.org/2001/XMLSchema#double
Resultado Comunidad de Madrid Investigación y Universidad 4.1539500000000333^http://www.w3.org/2001/XMLSchema#double
Resultado Andalucía Empresa Infraestructura Crítica 1.5088000000000572^http://www.w3.org/2001/XMLSchema#double
Resultado Aragón Empresa PYME Estándar 2.96400000000002704^http://www.w3.org/2001/XMLSchema#double
Resultado Comunidad Foral de Navarra Empresa Grande Estándar 1.902512500000016^http://www.w3.org/2001/XMLSchema#double
Resultado Castilla y León Empresa Infraestructura Crítica 2.150400000000058^http://www.w3.org/2001/XMLSchema#double
Resultado Extremadura Ciudadano Avanzado 3.6337500000001084^http://www.w3.org/2001/XMLSchema#double
Resultado Extremadura Empresa Infraestructura Crítica 3.9015000000001163^http://www.w3.org/2001/XMLSchema#double
Resultado Principado de Asturias Investigación y Universidad 1.7045333333333574^http://www.w3.org/2001/XMLSchema#double
Resultado Ceuta Empresa PYME Estándar 1.4079000000001118^http://www.w3.org/2001/XMLSchema#double
```

Figura 7: Resultados de datos obtenidos de la ontología

Una vez que obtenemos estos datos de las odontologías el siguiente paso es crear un programa de visualización mediante el uso de librerías JavaScript. Ya que se trata de la obtención de los riesgos que puede haber en las diferentes comunidades autónomas, se ha empleado la librería JQuery, en concreto empleando el plugin *maphilight* (11) mediante al cual permite añadir gráficos visuales a una imagen de un mapa usando *canvas* o *VML*.

### 3.2. REQUISITOS FUNCIONALES

Una vez que tenemos las etapas de diseño ya definidas, se explicará los requisitos funcionales de cada una de ellas ya que no en todas se obtiene el mismo tipo de datos o se realizan las mismas representaciones.

En esta primera parte del diseño, debido a que se emplean ontologías para la realización del razonamiento sobre amenazas empleando reglas SWRL, los resultados que se obtienen son resultados ontológicos, los cuales se pueden visualizar por medio de la herramienta *Protégé* (12).

Para realizar el razonamiento de los datos, se parte de una ontología limpia, vacía, que contiene el esqueleto del sistema pero que no hay ningún tipo de información de la configuración llamada “*Modric2.owl*”. Con el objetivo de conservar esa información para poder utilizarla si se desea cargar otra información se crea una copia de esta ontología con el nombre de “*Modric2Mirror.owl*”. Es sobre esta ontología sobre la que se introduce la información necesaria junto con las reglas SWRL para realizar el cálculo numérico del nivel de riesgo. Cada vez que se realiza un cálculo sobre amenazas se crea una copia de la ontología configurada anteriormente en una nueva ontología denominada “*Modric2Amenazas.owl*” (5) (6).

Finalmente, se va a trabajar sobre esta última copia ya que una vez hechos los cálculos se guarda la información y se elimina para que al analizar nuevas amenazas ya no existan las antiguas ya que si no la ontología tendría más individuos y amenazas y no realizaría correctamente los cálculos con el paso del tiempo.

En cuanto a la segunda etapa de diseño, se emplea la herramienta Apache Jena (13), en Eclipse, para la realización de la obtención de datos de la ontología. Una vez que se obtienen estos datos se realiza un guardado de los mismos en ficheros JSON para su posterior utilización en la parte de visualización. Cabe destacar que todos estos ficheros se guardan conjuntamente en una misma carpeta dentro de la parte de visualización ya que, en la última etapa del diseño se necesitará acceder a los mismos para realizar la representación sobre el mapa de España.

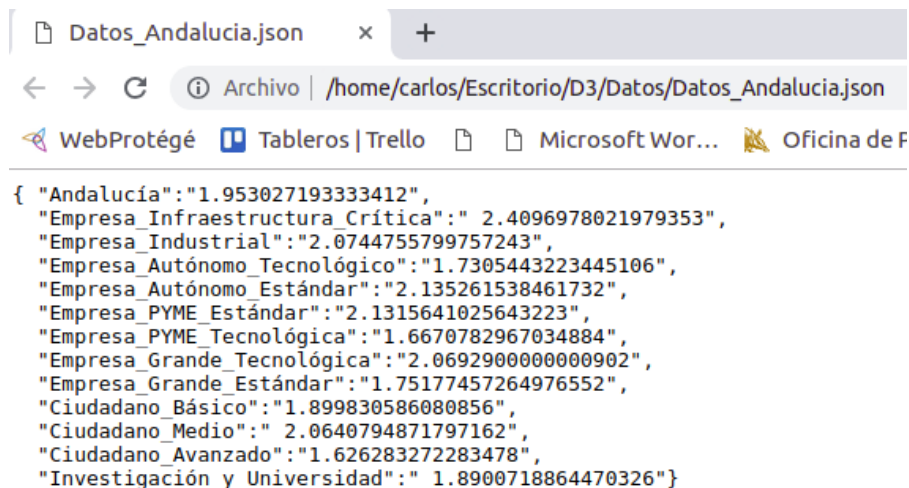
Este guardado de datos se realiza de manera individual, es decir, comunidad por comunidad de manera que cada Comunidad Autónoma tenga en su fichero, en primer lugar, el resultado del nivel de riesgo que hay en la comunidad y después los resultados de los diferentes niveles de riesgo que tienen los activos en dicha comunidad.

Existen dos maneras de poder visualizar estos datos que se han guardado en los ficheros JSON. La primera de ellas es directamente en la consola de ejecución de Eclipse, como se puede observar en la Figura 7. El problema que se encuentra al visualizar los resultados de esta manera es que el valor correspondiente a todos los activos que hay entre todas las Comunidades Autónomas, aparecen de manera desordenada y muy poco legible ya que tampoco siguen ningún orden lógico. Por tanto, la mejor manera de poder visualizar estos resultados es acceder a la carpeta donde guardamos todos los ficheros y al tratarse de ficheros JSON se podrá visualizar en un navegador Web sin necesidad de la utilización de un programa especial. Esto también presenta un problema y es que mediante este método solo vas a poder ver una serie de valores correspondientes a los resultados obtenidos, como se puede ver en la Figura 8. Sin embargo, el objetivo del trabajo es poder representar, en tiempo real, el análisis de las amenazas sobre una interfaz gráfica que en este caso se corresponde a la visualización sobre un mapa de España de los diferentes niveles de riesgo diferenciado por Comunidades Autónomas.

De esta manera llegamos a la tercera parte del diseño y consiste en la visualización de los datos. En esta parte se emplea la herramienta WebStorm (14) para la creación del programa que realizará la representación de los valores anteriormente obtenidos mediante el uso de HTML (15) debido a que permite dibujar, sobre un mapa de España, las diferentes Comunidades Autónomas y realizar interacciones sobre el mismo para que el usuario pueda tener una mejor perspectiva acerca de los resultados generados. Estas interacciones consisten, en primer lugar, representar cada comunidad con un color distinto dependiendo del nivel de riesgo que haya en esa comunidad (ver Figura 23). En segundo lugar, el usuario puede ver no solo el nivel de riesgo de la comunidad representado sobre el mapa sino también los niveles de riesgo de los diferentes activos, pinchando sobre la Comunidad Autónoma sobre la que quiere saber estos resultados y aparecerá una nueva página en la cual se representan los valores de riesgo de los activos sobre una tabla tal y como se puede observar en la Figura 31.

El objetivo principal del trabajo es que esta visualización se realice en tiempo real, por lo tanto, los datos tendrán que estar actualizándose constantemente o cada cierto intervalo de tiempo. Esta actualización de datos se consigue desde un primer momento con el razonamiento de los niveles de riesgo debido a que el programa que permite realizar el razonamiento, lo hace en *near-real time* (5) (6). Esto permite que la ontología de la que obtenemos los valores actualice sus datos en tiempo real, por lo tanto, la lectura de estos datos por parte del programa desarrollado en este trabajo tiene que realizarse también en tiempo real. Esto se consigue mediante el empleo de un tiempo de guardado o actualización de datos y un bucle *while* que permite que el programa este continuamente leyendo la ontología y cada cierto tiempo, cuando el usuario lo desee, se guarden los datos en los ficheros JSON correspondientes. Como la visualización realiza una lectura de estos ficheros para su representación, esta lectura también se está realizando continuamente, por lo tanto, la representación final cumplirá con el objetivo principal de desarrollar un sistema de visualización en tiempo real de los riesgos de ciberseguridad.

Esto se explica más detalladamente en los apartados 4.2 y 4.3 del documento.



```
{
  "Andalucía": "1.953027193333412",
  "Empresa_Infraestructura_Crítica": "2.4096978021979353",
  "Empresa_Industrial": "2.0744755799757243",
  "Empresa_Autónomo_Tecnológico": "1.7305443223445106",
  "Empresa_Autónomo_Estándar": "2.135261538461732",
  "Empresa_PYME_Estándar": "2.1315641025643223",
  "Empresa_PYME_Tecnológica": "1.6670782967034884",
  "Empresa_Grande_Tecnológica": "2.0692900000000902",
  "Empresa_Grande_Estándar": "1.75177457264976552",
  "Ciudadano_Básico": "1.899830586080856",
  "Ciudadano_Medio": "2.0640794871797162",
  "Ciudadano_Avanzado": "1.626283272283478",
  "Investigación_y_Universidad": "1.8900718864470326"}
}
```

**Figura 8: Resultado fichero Datos\_Andalucia.json**

## 4. DESARROLLO

Como hemos explicado en los puntos anteriores el desarrollo del sistema va a consistir en tres partes fundamentales de las cuales solo las dos últimas (Obtención de datos y Visualización) se explicarán de forma más detallada debido a que la primera parte (Razonamiento de las amenazas) consiste en un programa ya desarrollado (5) (6) pero esencial para el cálculo del riesgo sobre las amenazas en tiempo real.

### 4.1. RAZONAMIENTO DE LAS AMENAZAS

Consiste en un programa funcional cuya herramienta consiste en el razonamiento del riesgo de forma dinámica a partir de la entrada en el sistema de unos activos en forma de amenazas detectadas. De esta manera el cálculo del riesgo se puede realizar a distintos niveles de precisión (5):

- Riesgo global de España
- Riesgo de un activo específico en España
- Riesgo global de una Comunidad Autónoma
- Riesgo de un activo específico en una Comunidad Autónoma

Para el desarrollo de nuestro sistema de visualización nos interesa principalmente el cálculo del riesgo de los dos últimos niveles ya que se va a representar sobre un mapa de España solo el riesgo de las Comunidades Autónomas y sus activos.

Para la realización del cálculo del riesgo es necesario la intervención de una gran cantidad de subsistemas de los cuales solo nos centraremos en los que nos resultan de gran utilidad para nuestro proyecto que son: la configuración inicial, las ontologías y reglas de razonamiento y el razonador.

---

#### 4.1.1. CONFIGURACIÓN INICIAL

Mediante esta configuración inicial se obtiene los atributos específicos de los activos del archivo de la configuración. Este archivo es el principal para la realización del correcto cálculo de riesgos. Hay que tener en cuenta una serie de atributos (5) (6):

- **Tiempo de refresco de la consola:** Es el tiempo en segundos que indica cada cuanto tiempo se refresca el terminal de Linux con las amenazas entrantes y el tiempo transcurrido.
- **Número de amenazas de lanzamiento:** Es el número de amenazas entrantes que se han de detectar antes de que se realizado el cálculo del riesgo.
- **Tiempo de lanzamiento:** Es el tiempo, en segundos, que el sistema espera para realizar el cálculo del riesgo independientemente del número de amenazas entrantes.
- **Cabeceras:** Son las cabeceras con las cuales se clasifican las amenazas desde el INCIBE en formato CSV.
- **Tiempo de olvido:** Es el tiempo, en minutos, que define la antigüedad máxima de las mediciones para tener en cuenta para la realización del cálculo de riesgo.

- **Probabilidad:** Hace referencia a la probabilidad de que las amenazas que entran puedan afectar a los sistemas operativos en el hipotético caso de que no haya información sobre las mismas.
- **Confianza:** Hace referencia a la confianza de la fuente de información de la que se reciben las amenazas.
- **Correlación:** Modela la transferencia de riesgo entre comunidades autónomas.
- **Conocimiento:** Indica el conocimiento que tiene el activo en referencia a la ciberseguridad en España.
- **Peso de los activos:** Modela el peso que tiene cada individuo respecto al resto
- **Peso de los dispositivos:** Se corresponde con los pesos de cada dispositivo para cada individuo concreto.
- **Sistema operativo de los dispositivos:** Representa la penetración de cada sistema operativo por tipo de dispositivo y por individuo. Solo se da en los tipos de dispositivos móviles, PCs y servidores.

---

#### 4.1.2. ONTOLOGÍAS Y REGLAS SWRL

Una vez cargada la configuración inicial el siguiente paso es la introducción que contiene dicha configuración en la ontología usando el lenguaje OWL2. En primer lugar, se crea una ontología limpia la cual contiene el esqueleto del sistema y que todavía no contiene ningún tipo de información. Con el objetivo de conservar intacta toda la información se realiza una copia de esta ontología inicial sobre la cual se trabajará. Por tanto, es en esta copia en la se añade la información de la configuración inicial (5) (6).

Para realizar el cálculo numérico de las amenazas a través de un razonador semántico, se introducen reglas SWRL a través de las cuales se realiza un filtrado y un cálculo preliminar. Solo la amenaza que cumple con todas las condiciones es ejecutada por la regla. Hay que tener en cuenta que para el cálculo se tiene en cuenta parámetros introducidos en la configuración inicial para la definición de los activos e información propia de la amenaza (5) (6).

Por último, cada vez que se realiza un cálculo sobre amenazas se crea una tercera copia de la ontología configurada que es sobre la cual nosotros trabajaremos: *Modric2Amenazas.owl* (5) (6).

Un ejemplo de ontología se puede ver en la Figura 9.

---

#### 4.1.3. RAZONADOR

El modelado de las amenazas se realiza mediante las ontologías mientras que el análisis de riesgo se realiza mediante las reglas SWRL. Mas concretamente el razonador encargado de ejecutar las reglas y de comprobar la consistencia de la amenaza y por tanto el cálculo del riesgo es Pellet (5) (6).

Cabe destacar que el cálculo del riesgo de las Comunidades Autónomas se calcula a partir de los riesgos de los individuos que hay para cada comunidad y a su vez el riesgo que pueden sufrir estos individuos se calcula teniendo en cuenta el riesgo de los dispositivos que poseen y la importancia que tiene cada dispositivo para los individuos (5) (6).



Una vez que se conoce el riesgo de cada individuo o activo por Comunidad Autónoma el siguiente paso es conocer el riesgo de cada comunidad autónoma en su conjunto mediante la suma ponderada de los individuos de dicha comunidad por su peso (5) (6).

En el caso de que queramos conocer el riesgo a nivel nacional, éste se calcula mediante la suma ponderada del riesgo de cada comunidad con el número de amenazas materializadas en su territorio respecto al número de amenazas totales (5) (6).

## 4.2. OBTENCIÓN DE DATOS

Esta segunda parte del sistema consiste en la creación de un programa mediante el framework de Java, Apache Jena (16), el cual permite leer la ontología en la cual se ha realizado el razonamiento del riesgo tanto de las comunidades autónomas como de los diferentes activos que hay en cada comunidad. Antes de describir como es el programa que permite la lectura y obtención de los diferentes riesgos, es necesario analizar la ontología donde se ha realizado el razonamiento y a partir de la cual vamos a obtener los datos. Este análisis se realiza mediante el editor de ontologías Protégé (12) y cuyo resultado podemos ver a continuación:

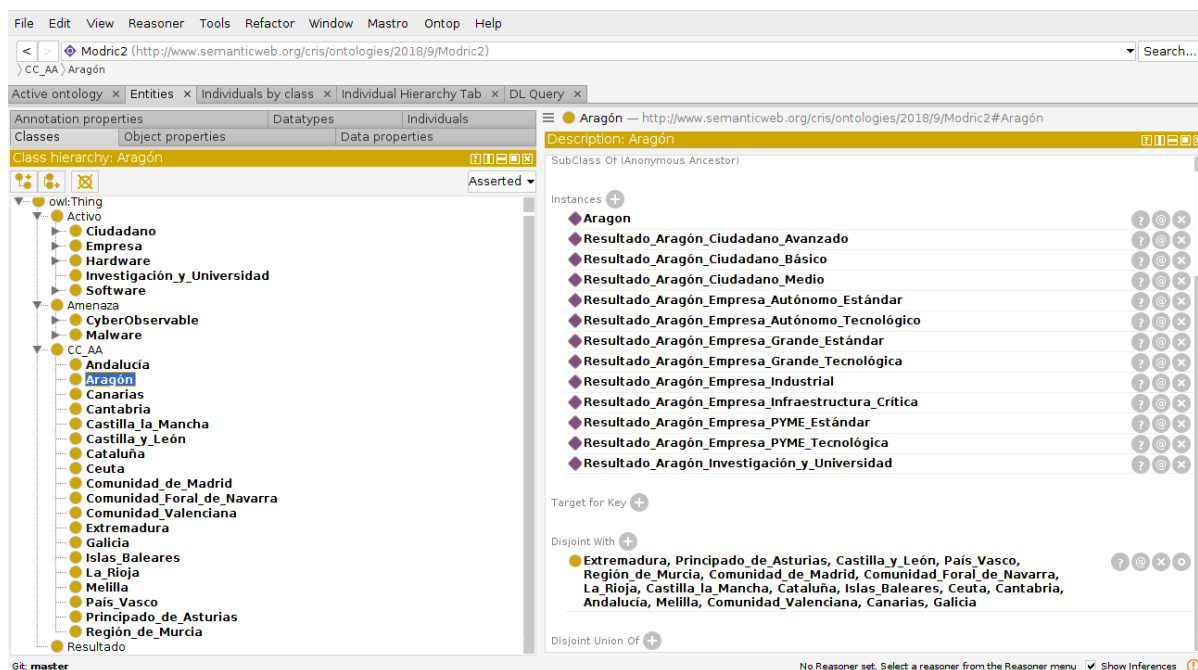


Figura 9: Resultado del análisis de la ontología

En esta imagen se pueden observar dos pantallas. La pantalla de la izquierda se muestra las clases principales que hay: Activo, Amenazas, CCAA (Comunidades Autónomas) y Resultados. De estas cuatro clases principales la única que nos importa y es la que contiene los resultados del riesgo calculado es la de CCAA. Dentro de ella se puede observar que se encuentran todas las comunidades autónomas que hay en España. Como hemos dicho en los puntos anteriores, no solo vamos a calcular el riesgo de una Comunidad Autónoma sino también el riesgo de sus activos. Para conocer sus activos solo es necesario pinchar en la comunidad que se desee y aparecerá la ventana de la derecha en la cual se encuentran todos los activos que posee dicha comunidad, en este caso se corresponde con Aragón.

Una vez que sabemos cuáles son los resultados de los activos solo falta por conocer cuál es el riesgo calculado. Al igual que hemos hecho antes con Aragón, solo se necesita pinchar en el resultado que se desee y ya se obtendría el valor deseado:

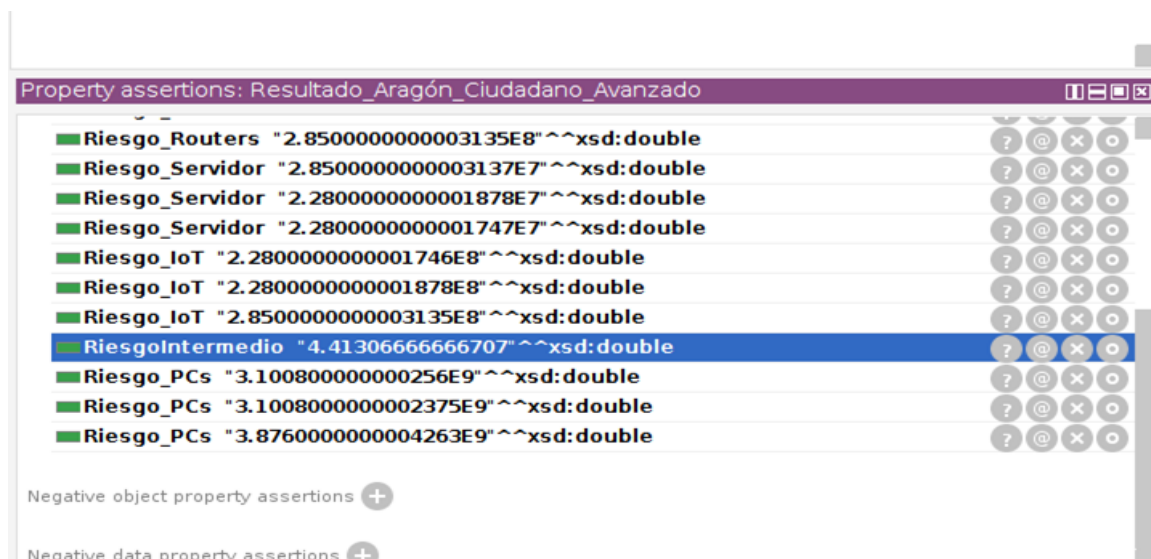


Figura 10: Resultado de un activo de Aragón

De todos los datos que aparecen el único que nos interesa es el *RiesgoIntermedio* el cual contiene el verdadero valor del riesgo calculado para ese activo.

#### 4.2.1. CONFIGURACIÓN DEL SISTEMA

Una vez que ya tenemos identificadas la clase a partir de la cual vamos a obtener los valores deseados, el siguiente paso es la realización de programa de extracción de dichos datos mediante Apache Jena. Para poder guardar los datos en tiempo real, es decir, que el programa este continuamente en funcionamiento sin que se pare al menos que el usuario lo desee, es necesario, en primer lugar, la creación de un fichero de configuración (Configuracion.txt) cuyo contenido es el siguiente:

```
#####
#Configuracion de la consola. Se puede configurar la frecuencia de refresco de la consola.Se puede configurar el tiempo de actualización de
datos
#####
#Cada cuantos segundos se refresca la consola.
TIEMPO.REFRESCO.CONSOLE = 2
#Cada cuantos segundos se actualizan los datos.
TIEMPO.LANZAMIENTO = 900
#####
```

Figura 11: Archivo Configuracion.txt

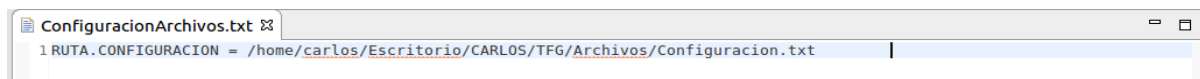
Este archivo es el encargado de la configuración de la consola. En él se pueden observar dos parámetros:

- **Tiempo de refresco de la consola:** Es el tiempo, en segundos, cada cuanto se refresca el terminal del programa para la realización del guardado de datos.
- **Tiempo de lanzamiento:** Es el tiempo, en segundos, cada cuanto se vuelven a guardar los datos, es decir, el tiempo que tarda en volverse a actualizar los datos extraídos de la ontología. Este tiempo puede variar desde 1 hasta el valor que se desee esperar a la



actualización. Si se desea que dicha actualización sea prácticamente en tiempo real este valor tiene que ser lo más próximo a 1, aunque se aconseja que sea de 5-10 ya que el programa de guardado tarda unos segundos en guardar todos los datos. Por lo tanto si el tiempo es muy pequeño, no dará tiempo a que el programa actualice correctamente los nuevos valores de riesgo obtenidos.

Para poder acceder a este fichero desde el programa de Eclipse, es necesario crear un archivo denominado *ConfiguraciónArchivos.txt* el cual contiene la ruta donde se encuentra el fichero anteriormente descrito:



**Figura 12: Ruta del archivo de Configuración.txt**

Una vez que ya tenemos la dirección en la que se encuentra nuestro fichero de configuración, creamos dos clases:

- **ConfigurationFilePath.java:** Esta clase nos permite acceder al fichero *Configuration.txt* y cargar las propiedades que tiene. En Este caso la única propiedad que tiene el fichero es la ruta en la que se encuentra el archivo de configuración y cuyo nombre es RUTA.CONFIGURACIÓN. Este valor lo guardamos en una tabla hash (17) con una clave y un valor ya que en el caso de que existan otros archivos a los que haya que acceder, esta es la manera más eficaz de poder obtener los valores de dichos archivos.

Cabe destacar que al disponer solo de una ruta de configuración y no existen problemas de que se solape o se confunda con otras rutas, se puede poner la ruta donde se encuentra el fichero de configuración directamente en la clase de Eclipse, pero se ha realizado de esta manera con la visión de posibles mejoras futuras.

- **Configuration.java:** Esta clase es muy parecida a su predecesora con la diferencia es que esta clase es la que nos permite leer y obtener los valores que hay en el fichero *Configuration.txt*. Para poder obtener estos valores primero es necesario llamar a la clase *ConfigurationFilePath.java* mediante una clave a partir de la cual, al estar asociada mediante la relación clave-valor, podemos obtener su valor asociado que en este caso es la ruta de configuración.

Una vez que ya conocemos la ruta donde se encuentran los valores deseados, creamos dos métodos, uno para obtener el tiempo de refresco de la consola y el otro para obtener el tiempo de lanzamiento y lo guardamos en dos variables para su posterior utilización (*tiempoRefrescoConsola* y *tiempoTrigger*, respectivamente).

```

1 package TFG;
2
3 import java.io.FileInputStream;
4
5
6
7
8
9 public class ConfigurationFilesPath {
10     public static final String rutaArchivoConfiguracionArchivos = "./ConfiguracionArchivos.txt";
11     public static HashMap<String, String> getFilesPath() {
12         Properties prop = new Properties();
13         InputStream is = null;
14
15         try {
16             is = new FileInputStream(rutaArchivoConfiguracionArchivos);
17             prop.load(is);
18         } catch (IOException e) {
19             System.out.println(e.toString());
20         }
21
22         HashMap<String, String> rutasArchivos = new HashMap<String, String>();
23         rutasArchivos.put("configuracion", prop.getProperty("RUTA.CONFIGURACION"));
24         return rutasArchivos;
25     }
26     public static void main (String [] args) {
27         getFilesPath();
28     }
29 }
30

```

Figura 13: Clase ConfigurationFilesPath.java

```

1
2
3
4
5
6
7 public class Configuration {
8
9
10     public static final String rutaArchivoConfiguracion = ConfigurationFilesPath.getFilesPath().get("configuracion");
11     public static int tiempoTrigger;
12     public static int tiempoRefrescoConsola;
13
14
15     public static int getTiempoRefrescoConsola() {
16         Properties prop = new Properties();
17         InputStream is = null;
18
19         try {
20             is = new FileInputStream(rutaArchivoConfiguracion);
21             prop.load(is);
22         } catch (IOException e) {
23             System.out.println(e.toString());
24         }
25
26         return tiempoRefrescoConsola = Integer.parseInt((prop.getProperty("TIEMPO.REFRESCO.CONSOLA").replace(" ", "")));
27     }
28
29     public static int getTiempoTrigger() {
30         Properties prop = new Properties();
31         InputStream is = null;
32
33         try {
34             is = new FileInputStream(rutaArchivoConfiguracion);
35             prop.load(is);
36         } catch (IOException e) {
37             System.out.println(e.toString());
38         }
39
40         return tiempoTrigger = Integer.parseInt((prop.getProperty("TIEMPO.LANZAMIENTO").replace(" ", "")));
41     }
42
43     public static void main(String []args) {
44         getTiempoRefrescoConsola();
45         getTiempoTrigger();
46     }
47 }
48
49

```

Figura 14: Clase Configuration.java

Lo que nos permite estas dos clases es establecer el guardado de los resultados en *near-real time*, eso es, guardar tanto el riesgo de cada comunidad como el de sus activos adjuntos en ficheros JSON cada cierto tiempo, dependiendo del tiempo de lanzamiento que establezcamos siempre teniendo en cuenta que el programa va a estar funcionando. Por lo tanto, en el programa principal, antes de obtener los resultados de la ontología y guardar dichos resultados en sus ficheros correspondientes, es necesario llamar a las clases antes descritas para obtener el valor del tiempo de guardado.

#### 4.2.2. GUARDADO Y ACTUALIZACIÓN DE DATOS

Como hemos dicho anteriormente cada resultado se guarda en sus ficheros correspondientes, es decir, los datos de cada comunidad autónoma se van a guardar en ficheros distintos con el objetivo final de facilitar la lectura de datos para la visualización final.

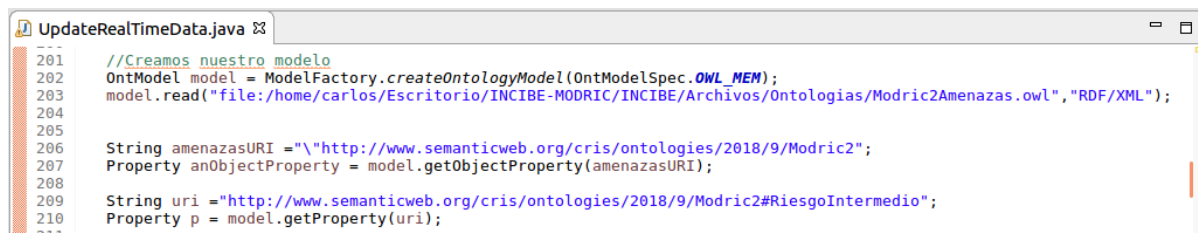
El guardado de datos se realiza siguiendo los siguientes pasos. Todos los pasos se realizan dentro de una clase principal denominada *UpdateRealTimeData*:

- 1) **Creación del modelo ontológico:** Lo primero que hay que hacer es crear el modelo ontológico el cual nos permite tener acceso a la ontología que contiene el cálculo del riesgo realizado y obtener el valor de dicho riesgo como una propiedad de la ontología.

Creamos la variable *model* como un nuevo modelo de ontología que responde a unas especificaciones y que incluye un *ModelMarker* que crea el modelo básico necesario.

Para poder leer este modelo empleamos la propiedad *read* la cual, a partir de la dirección donde se encuentra la ontología con los valores y el idioma de serialización que para la ontología creada es *RDF/XML*, obtenemos un modelo ontológico al cual ya podemos obtener de él las propiedades deseadas. Como hemos explicado en el punto 4.1.2, los resultados finales del razonamiento de amenazas se guardan en una ontología cuyo nombre es “Modric2Amenzas” (5) (6), por consiguiente, esta será la ontología a partir de la cual obtengamos los riesgos deseados.

Al tratarse de un modelo RDF, podemos obtener de él varias propiedades. Como explicamos en la introducción del punto 4.2, el único valor que nos interesa al examinar la ontología es el valor correspondiente al *RiesgoIntermedio*, el cual es una propiedad del modelo ontológico y cuyo valor se obtiene mediante la propiedad “getProperty (String)” (13). El resultado que devuelve es una propiedad RDF. El parámetro *String* se corresponder con la URI de la ontología especificada para los valores correspondientes al *RiesgoIntermedio* mediante la siguiente notación: <http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#RiesgoIntermedio>.



```

201 //Creamos nuestro modelo
202 OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
203 model.read("file:/home/carlos/Escritorio/INCIBE-MODRIC/INCIBE/Archivos/Ontologias/Modric2Amenzas.owl", "RDF/XML");
204
205
206 String amenazasURI = "http://www.semanticweb.org/cris/ontologies/2018/9/Modric2";
207 Property anObjectProperty = model.getObjectProperty(amenazasURI);
208
209 String uri = "http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#RiesgoIntermedio";
210 Property p = model.getProperty(uri);
211

```

Figura 15: Creación del modelo ontológico. Clase *UpdateRealTimeData.java*

- 2) **Recorrido de la ontología:** Antes de empezar a recorrer la ontología para poder guardar los datos de esta, es necesario crear una serie de arrays para almacenar dichos datos. Como cada comunidad va a guardar sus propios datos, vamos a crear tantos arrays como Comunidades Autónomas haya. Al haber dos tipos de datos que nos interesan, a su vez, por comunidad, se crea dos listas de arrays distintos, uno que contenga el nombre local correspondiente a los activos que tenga cada comunidad y otro con el resultado del riesgo de cada activo. El nombre correspondiente a cada array tendrá la siguiente forma: *collectionIndNombreComunidad* para guardar el nombre de los activos por comunidad y *collectionIterStatementNombreComunidad* para guardar el resultado correspondiente a cada activo.

Por otro lado, para poder guardar el riesgo razonado por comunidad, éste al encontrarse dentro de la ontología en una clase distinta a la que se encuentran los activos, necesitamos también dos arrays distintos uno para guardar el nombre de la Comunidad Autónoma cuyo nombre será

*collectionInd* y otro para guardar el resultado del riesgo asociado a cada una de ellas, cuyo nombre será *collectionIterStatement*.

Para poder recorrer el modelo ontológico creado previamente y obtener los datos anteriormente descritos, es necesario crear dos bucles *for*, el primero para recorrer las clases principales de las ontologías y el segundo para recorrer las instancias que tiene cada clase principal. De esta manera, mediante el segundo bucle, vamos a obtener una serie de datos que no resultan muy relevantes para nuestro proyecto por lo tanto solo necesitamos conocer las propiedades de las instancias que contengan los resultados del riesgo analizado y eso se consigue mediante la propiedad “listProperties(p)” (13) cuyo resultado se guarda en un iterador que devuelve declaraciones RDF. El valor del argumento *p* se corresponde con el valor de la propiedad RDF anteriormente descrita.

Finalmente, mediante el empleo de sentencias *if*, una por comunidad, guardamos los resultados de la ontología en sus correspondientes arrays. La condición de cada sentencia responde de forma verdadera si los recursos se corresponden con a la clase indicada por el URI dado.

Existe un URI por Comunidad Autónoma, para obtener los valores de los activos (<http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#NombreComunidad>) y un URI correspondiente a la clase principal CCAA, que contiene el nombre de todas las Comunidades Autónomas ([http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#CC\\_AA](http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#CC_AA)).

Al finalizar estos pasos ya tenemos guardados en los arrays todos los datos que posteriormente vamos a representar. El siguiente paso es el guardado de los resultados en ficheros JSON.



```

222 //Recorremos la ontologia
223 for (Iterator<OntClass> i = model.listHierarchyRootClasses(); i.hasNext();){
224
225     OntClass cls = i.next();
226     for (Iterator it = cls.listInstances(true); it.hasNext();){
227
228         Individual ind = (Individual)it.next();
229         StmtIterator iterStatement = ind.listProperties(p);
230
231         //CCAA
232         if(ind.hasRDFTYPE("http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#CC_AA")) {
233
234             collectionInd.add(ind.getLocalName());
235
236             String amenaza= iterStatement.nextStatement().getObject().toString();
237             collectionIterStatement.add(amenaza);
238
239         }
240
241         //Andalucia
242         if(ind.hasRDFTYPE("http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#Andalucia")) {
243
244             collectionIndAndalucia.add(ind.getLocalName());
245
246             String amenazaAndalucia = iterStatement.nextStatement().getObject().toString();
247             collectionIterStatementAndalucia.add(amenazaAndalucia);
248
249         }
250
251         //Aragon
252         if(ind.hasRDFTYPE("http://www.semanticweb.org/cris/ontologies/2018/9/Modric2#Aragón")) {
253
254             collectionIndAragon.add(ind.getLocalName());
255
256             String amenazaAragon = iterStatement.nextStatement().getObject().toString();
257             collectionIterStatementAragon.add(amenazaAragon);
258
259         }
260
261     }
262 }

```

**Figura 16: Recorrido de la ontología: Clase UpdateRealTimeData.java**

- 3) **Guardado de datos en ficheros JSON:** Como se ha explicado anteriormente el guardado de datos se va a realizar en ficheros JSON, de esta manera, la lectura posterior de los datos para su visualización se realizará de forma más sencilla.

Al igual que ocurre para obtener los datos de la ontología, necesitamos crear objetos y arrays JSON tantos como Comunidades Autónomas haya. Ambos nos van a permitir guardar los datos siguiendo el modelo JSON ya que mediante la clase *JSONObject* (18) representamos un objeto

JSON inmutable, la clase *JSONArray()* (19) representa un array JSON inmutable el cual creamos para guardar los valores de los arrays creados previamente.

Para poder escribir los datos en un fichero JSON, se hace uso de la clase *FileWriter* (20). Se van a crear tantos objetos de esta clase como número de Comunidades Autónomas existan. Por comunidad, se va a escribir la ruta donde queramos guardar los ficheros. Esta ruta será una carpeta, la cual el programa de visualización accederá para poder representar el riesgo.

Hay que tener en cuenta que, si esta ruta no existe, se crea automáticamente. En caso contrario sobrescribe los datos guardado previamente en esa ruta. Finalmente se va a escribir el texto correspondiente en el fichero mediante el método *write* (20) el cual utiliza como parámetro un *String*.



```

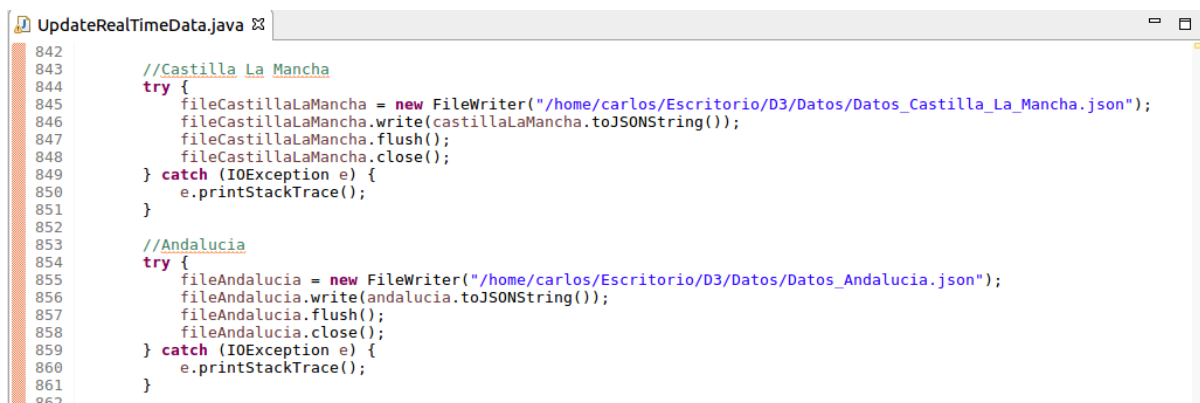
474 // Meter datos en Castilla la Mancha
475
476 castillaLaMancha.put(collectionInd.get(0), collectionIterStatement.get(0));
477
478 for(int clm=0; clm<collectionIndCastillaLaMancha.size(); clm++) {
479     listIndCastillaLaMancha.add(collectionIndCastillaLaMancha.get(clm));
480 }
481 for(int clm1=0; clm1<collectionIterStatementCastillaLaMancha.size(); clm1++) {
482     listIterStatementCastillaLaMancha.add(collectionIterStatementCastillaLaMancha.get(clm1));
483 }
484
485 for(int clm2=0; clm2<listIndCastillaLaMancha.size(); clm2++) {
486     for(int clm3=0; clm3<listIterStatementCastillaLaMancha.size(); clm3++) {
487         castillaLaMancha.put(listIndCastillaLaMancha.get(clm2), listIterStatementCastillaLaMancha.get(clm3));
488     }
489 }
490
491 // Meter datos en Andalucía
492
493 andalucia.put(collectionInd.get(1), collectionIterStatement.get(1));
494
495 for(int and=0; and<collectionIndAndalucia.size(); and++) {
496     listIndAndalucia.add(collectionIndAndalucia.get(and));
497 }
498 for(int and1=0; and1<collectionIterStatementAndalucia.size(); and1++) {
499     listIterStatementAndalucia.add(collectionIterStatementAndalucia.get(and1));
500 }
501
502 for(int and2=0; and2<listIndAndalucia.size(); and2++) {
503     for(int and3=0; and3<listIterStatementAndalucia.size(); and3++) {
504         andalucia.put(listIndAndalucia.get(and2), listIterStatementAndalucia.get(and3));
505     }
506 }
507
508

```

**Figura 17: Guardado de fichero. Clase UpdateRealTimeData.java**

- 4) **Actualización de los datos:** Una vez que ya sabemos cómo guardar los resultados del análisis de riesgo, habrá que actualizarlos cada cierto tiempo. Como hemos dicho anteriormente, la ruta donde se guardan los ficheros se sobrescribe cada vez que se vuelven a guardar nuevos datos, por lo tanto, para actualizar, solo es necesario volver a realizar los pasos 2 y 3 el tiempo que deseamos.

Para que el programa esté funcionando continuamente sin detenerse, los pasos 2 y 3 se realizarán dentro de un bucle *while* el cual siempre se cumple. A su vez el paso 2 estará dentro de otro bucle *while* interior el cual solo estará ejecutándose durante el tiempo de lanzamiento, es decir, hasta que no finalice el tiempo que habías programado previamente de cada cuanto tiempo se guardan los datos no pasa al siguiente paso. Si el usuario lo desea y no quiere esperar que se cumpla el tiempo de lanzamiento para guardar los datos, existe una hebra de lectura por consola, la cual permite salir de este bucle interior y seguir con el paso 3 siempre y cuando escribamos la letra *S* en dicha consola. De esta manera obtenemos los datos guardados y actualizados en *near-real time* cumpliéndose el objetivo principal del proyecto.



```

842
843 //Castilla La Mancha
844 try {
845     fileCastillaLaMancha = new FileWriter("/home/carlos/Escritorio/D3/Datos/Datos_Castilla-La_Mancha.json");
846     fileCastillaLaMancha.write(castillaLaMancha.toJSONString());
847     fileCastillaLaMancha.flush();
848     fileCastillaLaMancha.close();
849 } catch (IOException e) {
850     e.printStackTrace();
851 }
852
853 //Andalucia
854 try {
855     fileAndalucia = new FileWriter("/home/carlos/Escritorio/D3/Datos/Datos_Andalucia.json");
856     fileAndalucia.write(andalucia.toJSONString());
857     fileAndalucia.flush();
858     fileAndalucia.close();
859 } catch (IOException e) {
860     e.printStackTrace();
861 }
862

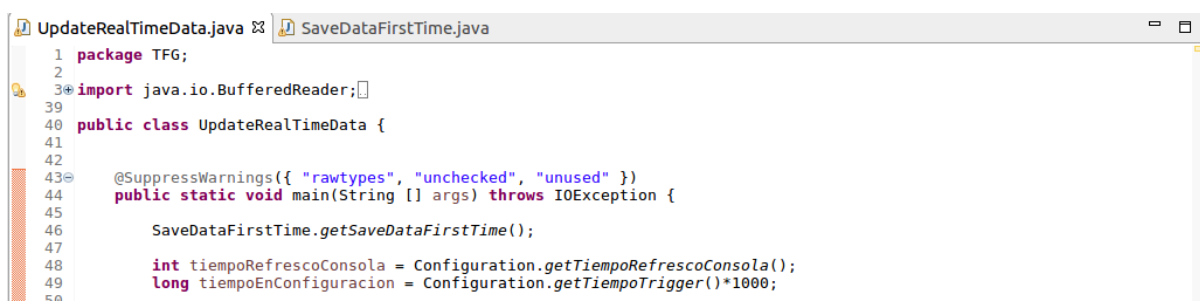
```

Figura 18: Actualización de datos. Clase UpdateRealTimeData.java

Realizando de esta manera el guardado de datos, cuando se ejecuta el programa por primera vez nos encontramos con un problema, principalmente para tiempos de lanzamiento muy grandes, el cual consiste en que los datos no se guardarían hasta que no se cumpliera ese tiempo o escribiéramos la letra *S* por consola.

Para poder suplir este problema se ha realizado una clase secundaria la cual tiene un solo método el cual ejecuta solo los pasos 1, 2 y 3. Esta clase recibe el nombre de *SaveDataFirstTime*.

Por lo tanto, en la clase principal, lo primero que se hace antes de realizar la configuración del sistema y el guardado y actualización de los datos, se llama a la clase secundaria para realizar un primer guardado y así tener datos al instante para poder representarlos. Una vez que se realiza este paso, se continua con todo lo descrito anteriormente.



```

1 package TFG;
2
3 import java.io.BufferedReader;
39
40 public class UpdateRealTimeData {
41
42
43 @SuppressWarnings({ "rawtypes", "unchecked", "unused" })
44 public static void main(String [] args) throws IOException {
45
46     SaveDataFirstTime.getSaveDataFirstTime();
47
48     int tiempoRefrescoConsola = Configuration.getTiempoRefrescoConsola();
49     long tiempoEnConfiguracion = Configuration.getTiempoTrigger()*1000;
50

```

Figura 19: Inicio Clase UpdateRealTimeData.java

Por último, cabe destacar que se ha creado un fichero el cual guarda la fecha y la hora de la última actualización o guardado de datos. Estos nos van a resultar útil en la visualización ya que de esta forma se puede conocer cuando ha sido la última vez que se ha actualizado el riesgo y tener una mejor visualización en tiempo real.

Este guardado también se realiza sobre un fichero JSON, pero de una manera distinta a los guardados anteriores. En primer lugar, una vez que se comprueba que se han guardado todos los datos correctamente, se crea una variable de tipo *String* la cual contendrá la fecha y la hora de guardado. Esto se consigue mediante el método *format* el cual transforma la fecha en tiempo actual de formato *Data* a un formato de tipo *String*. El siguiente paso es transformar esa variable creada a formato JSON mediante el método *toJson* (21) el cual serializa el objeto pasado como parámetro a su objeto JSON correspondiente, es decir, convierte objetos Java en objetos JSON.



Finalmente guardamos este dato en un fichero para su posterior utilización en la representación visual de la misma manera que hemos realizado el guardado de los datos anteriormente.

```
String fechaActual = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(new Date(System.currentTimeMillis()));
String f = gson.toJson(fechaActual);
String ultimaActualizacion = "Última actualización a las";
String a = gson.toJson(ultimaActualizacion);
String fe = "{" + a + ":" + f + "}";
try{
    FileWriter writer = new FileWriter("/home/carlos/Escritorio/D3/Datos/Fecha_ultima_Ejecucion.json");
    writer.write(fe);
    writer.close();
}catch (IOException e) {
    e.printStackTrace();
}
```

**Figura 20: Guardado fecha y hora de la última actualización**

### 4.3. VISUALIZACIÓN

La representación visual de los resultados del razonamiento de amenazas sobre un mapa de España se va a realizar mediante lenguaje HTML (15) el cual permite la creación de mapas iterativos en páginas web mediante el uso de plugins, más concretamente, *Maphiligh* (11) el cual permite añadir y crear efectos visuales a una imagen de un mapa usando *canvas* o *VML*.

Para poder realizar todo el desarrollo de la parte visual se ha empleado el IDE JavaScript WebStorm (14) el cual es compatible con el uso de una amplia gama de tecnologías modernas relacionadas con el lenguaje de programación JavaScript, HTML y CSS.

Para poder utilizar este plugin, lo primero que se debe de hacer es incluir las librerías de JQuery y el *JQuery Maphilight* el cual nos permite tener acceso al plugin:

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
<script type="text/javascript" src="jquery.maphilight.min.js"></script>
```

**Figura 21: Librería JQuery y JQuery Maphilight**

Una vez que ya hemos cargado las librerías para la representación completa del mapa se divide en dos partes: lectura de los datos de riesgo y representación de los datos sobre el mapa.

#### 4.3.1. LECTURA DE DATOS

Es la parte principal y previa a la representación de los datos sobre el mapa de España. Si esta lectura no se realiza correctamente, la representación de los datos se realizará de forma errónea y por tanto lo que estaremos viendo no se corresponderá con la realidad.

Al igual que ocurre con la obtención de datos en el punto 4.2 del documento, la representación que se tiene que realizar es tiempo real, es decir, los datos que se están representando están constantemente actualizando cada cierto tiempo. Esto se consigue haciendo una lectura de datos en la misma dirección donde se han guardado. Hay que destacar que esta lectura se tiene que realizar con cada comunidad a representar debido a que los datos del riesgo y sus activos correspondientes se guardan individualmente comunidad por comunidad.

La lectura es una función simple la cual tiene como objetivo acceder al archivo de datos especificado pasando como parámetro la dirección de guardado de los mismos. Una vez que haya encontrado el fichero JSON se realiza la lectura de los datos mediante el método *JSON.parse()* (22) el cual analiza la cadena de texto que se pasa como parámetro, en este caso una cadena de texto JSON y lo transforma a un valor que posteriormente será analizado. El análisis de estos valores nos servirá, en primer lugar, para representar el riesgo de la comunidad mediante un color

sobre el mapa y en segundo lugar para crear una tabla en la cual se podrá observar los activos de cada Comunidad Autónoma.

```
<script type="text/javascript">
    function read_Datos_Andalucia(file, callback) {
        var rawFile = new XMLHttpRequest();
        rawFile.overrideMimeType("application/json");
        rawFile.open("GET", file, true);
        rawFile.onreadystatechange = function() {
            if (rawFile.readyState === 4 && rawFile.status === "200") {
                callback(rawFile.responseText);
            }
        }
        rawFile.send(null);
    }

    read_Datos_Andalucia("/Datos/Datos_Andalucia.json", function(text){
        var data_andalucia = JSON.parse(text);
        representacion(data_andalucia);
    });
</script>
```

**Figura 22:Función de obtención de datos de Andalucía**

En cuanto se haya realizado las funciones de obtención de datos de todas las Comunidades Autónomas, el siguiente paso es la representación de estos.

#### 4.3.2. REPRESENTACIÓN DE LOS DATOS

La representación de los datos se va a realizar sobre un mapa de España dividido por Comunidades Autónomas. El objetivo es realizar una visualización en tiempo real del riesgo que existe en cada comunidad. Para cumplir este objetivo, la mejor manera de representar este riesgo es establecer una serie de niveles de riesgo, los cuales se representará cada uno por medio de un color. De esta manera, el usuario solo con ver el mapa sabrá el nivel de riesgo que tiene de sufrir una amenaza. Por otro lado, se permitirá al usuario saber el riesgo exacto que hay en cada comunidad, así como el riesgo por activo y cuyo resultado se muestra en una tabla para una mejor visualización. Para conseguir estos, solo será necesario pinchar sobre el mapa en la Comunidad Autónoma que se desee y se mostrará lo anteriormente descrito.

#### NIVELES DE RIESGO

Se van a establecer tres niveles de riesgo cada uno representado mediante un color. El riesgo más bajo se corresponde con el valor 0.0 (nunca se va a alcanzar este valor ya que por muy protegido que estén los equipos siempre va a ver una pequeña posibilidad de que estos sean atacados, por tanto, siempre va a ver un riesgo de sufrir una amenaza) mientras que el nivel más alto se corresponde con el valor de 10.0:

- **Nivel bajo (color verde):** Indica que el riesgo de amenaza de sufrir un ataque cibernético es realmente bajo lo que indica que la comunidad, así como los activos de la comunidad están bien protegidos. Hay que tener en cuenta que, aunque el nivel de riesgo sea bajo,



no hay que descuidar la protección de los equipos de las empresas ya que en cualquier momento se puede realizar un ataque que afecte a dichos equipos. A su vez dentro de este nivel existen otros tres tipos de niveles los cuales nos dan más información sobre el tipo de riesgo que hay.

Este nivel se corresponde con los valores de riesgo comprendidos entre 0.0-3.3.

- **Nivel medio (color amarillo):** Indica que el riesgo de amenaza de sufrir un ataque cibernético es moderado lo que indica que tanto los usuarios de los dispositivos electrónicos como las empresas o activos de la comunidad tiene que estar alerta ante un posible ataque que afecte a sus equipos. El daño de los ataques se puede ver reducido dependiendo del nivel de protección que disponga el equipo, por lo que para equipos que estén bien protegidos ya sea mediante sistemas de antivirus u otros mecanismos, el daño producido por el ataque será mucho menor que en equipos menos protegidos. Al igual que ocurre con el nivel bajo, dentro de este nivel también existen tres subniveles los cuales nos dan una información aún más exacta.

Este nivel se corresponde con los valores de riesgo comprendidos entre 3.4-6.6.

- **Nivel alto (color rojo):** Indica que el riesgo de amenaza de sufrir un ataque cibernético es realmente alto por lo que en cualquier instante de tiempo se puede producir dicho ataque. De esta manera tanto las comunidades afectadas como las empresas afectadas tendrán que estar preparadas a mitigar los efectos producidos por el ataque. Al igual que ocurre como en los anteriores niveles, los daños que puedan sufrir los equipos afectados se verán reducidos dependiendo del nivel de protección que dispongan. También existen tres subniveles para poder visualizar de manera más precisa el nivel de riesgo.

Este nivel se corresponde con los valores de riesgo comprendidos entre 6.7-10.0.

Nivel de riesgo	Color representación	Descripción	Rango valores	Subniveles	Color representación	Rango valores
Bajo	Verde	El riesgo de sufrir una amenaza es bajo o nulo	0.0-3.3	Bajo bajo		0.0-1.1
				Bajo medio		1.2-2.2
				Bajo alto		2.3-3.3
Medio	Amarillo	El riesgo de sufrir una amenaza es moderado	3.4-6.6	Medio bajo		3.4-4.4
				Medio medio		4.5-5.5
				Medio alto		5.6-6.6
Alto	Rojo	El riesgo de sufrir una amenaza es alto o extremo	6.7-10.0	Alto bajo		6.7-7.7
				Alto medio		7.8-8-8
				Alto alto		8.9-10.0

**Figura 23: Niveles de riesgo**

Hay que resaltar que estos niveles no nos indican si el ataque que se realiza sobre los equipos será muy grave o apenas afectará a los equipos. Lo que nos indica esta representación es el nivel de riesgo a sufrir un ataque sea del tipo que sea.

Como se lleva diciendo a lo largo del trabajo, la representación se va a realizar sobre un mapa de España dividido por Comunidades Autónomas. Para poder realizar este mapa, en primer lugar, se toma una imagen de un mapa y se realiza un mapeo de este, comunidad por comunidad. Así de esta manera obtenemos las coordenadas geográficas, dentro de la imagen, de todas las Comunidades Autónomas. Solo falta añadir el nivel de riesgo dependiendo de los datos leídos por comunidad.

Como se puede observar en la Figura 22, para añadir estos valores al mapa se llama a una función la cual es la encargada de leer, dentro del fichero JSON, el riesgo asociado a la comunidad, el cual

se encuentra como dato en la primera línea del fichero. La lectura de este dato y junto a los niveles de riesgo previamente establecidos coloreará la comunidad con el nivel de riesgo correspondiente.

A su vez para obtener más información sobre el riesgo de amenaza en una comunidad, ya que solo viendo el mapa sabemos el nivel de riesgo, pero no el riesgo exacto, se puede pinchar sobre la Comunidad Autónoma sobre la que se desea obtener más información y el resultado es una tabla en la cual se muestra el valor real del nivel riesgo, así como el valor de los riesgos de los diferentes activos que hay por comunidad divididos en: empresas, ciudadanos e investigación y universidad.

Por otro lado, para facilitar la visualización del riesgo, mediante código HTML se añade una leyenda parecida a la Figura 23, la cual nos permite, a simple vista, conocer entre que valores se encuentra el riesgo por comunidad.

Por último, se añade un cuadro el cual contiene información de cuando se ha realizado la última actualización de los datos. Esta información es la fecha y la hora de la última actualización. Para obtener estos datos y posteriormente representarlo se hace de una manera muy similar a la de obtención de los datos del cálculo del riesgo. En primer lugar, se crea una variable que accede al fichero donde se encuentran los datos que se van a representar. A continuación se crea una función `$.getJSON` la cual hace una petición de datos al fichero previamente creado y mediante el método `JSON.stringify()` (23) obtenemos el valor del contenido de fichero y posteriormente lo representamos mediante el método `document.write()`.

```
<script type="text/javascript" src="Datos/Fecha ultima Ejecucion.json">
  var file = 'file:///home/carlos/Escritorio/D3/Datos/Fecha ultima Ejecucion.json';

  var myGet = $.getJSON(file, function() {
  });

  myGet.done(function(data) {
    var content = JSON.stringify(data);
    document.write(content);
  });
</script>
```

**Figura 24: Obtención datos última actualización**

## 5. RESULTADOS

Una vez que empezamos la ejecución del programa vamos a tener tres tipos de resultados que se van a analizar:

1. **Resultados de los activos por comunidades autónomas:** Se corresponde con los datos extraídos de la ontología el cual nos permitirá conocer como guardar dichos datos en fichero JSON.
2. **Resultados del guardado de datos:** Se corresponde con los resultados una vez que se han guardado todos los datos en sus correspondientes ficheros JSON.
3. **Resultados de la visualización:** Se corresponde con los resultados de la visualización final en tiempo real.

### 5.1. RESULTADOS DE LOS ACTIVOS

Estos resultados se obtienen una vez que se ha recorrido la ontología que contiene los valores del riesgo razonado sobre las amenazas. Como hemos indicado en el apartado 4.2.2 del propio documento, se hará varios recorridos por la ontología. En primer lugar, para obtener el nombre de cada Comunidad Autónoma y el riesgo correspondiente a cada una de ellas y en segundo lugar se obtendrá el valor de los activos correspondientes a cada comunidad, obteniendo así los siguientes resultados:

```
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#noProviders for further details.
Castilla la Mancha 2.2214579200001254^http://www.w3.org/2001/XMLSchema#double
Andalucía 1.953027193333412^http://www.w3.org/2001/XMLSchema#double
La Rioja 1.6180879866667257^http://www.w3.org/2001/XMLSchema#double
Canarias 1.8194475600000548^http://www.w3.org/2001/XMLSchema#double
Melilla 1.2985720000000227^http://www.w3.org/2001/XMLSchema#double
Comunidad de Madrid 4.895350200000392^http://www.w3.org/2001/XMLSchema#double
Galicia 1.6451834000001146^http://www.w3.org/2001/XMLSchema#double
Comunidad Valenciana 1.4560059800001106^http://www.w3.org/2001/XMLSchema#double
Principado de Asturias 1.7834376533333582^http://www.w3.org/2001/XMLSchema#double
Castilla y León 3.0622696533334164^http://www.w3.org/2001/XMLSchema#double
Comunidad Foral de Navarra 1.606687250000135^http://www.w3.org/2001/XMLSchema#double
Aragón 3.4292509333336465^http://www.w3.org/2001/XMLSchema#double
Cantabria 1.0165658400000792^http://www.w3.org/2001/XMLSchema#double
Región de Murcia 1.714138300000035^http://www.w3.org/2001/XMLSchema#double
Islas Baleares 2.0270552800000647^http://www.w3.org/2001/XMLSchema#double
Cataluña 1.201664776666742^http://www.w3.org/2001/XMLSchema#double
País Vasco 1.68685626666673^http://www.w3.org/2001/XMLSchema#double
Extremadura 4.298449500000128^http://www.w3.org/2001/XMLSchema#double
Ceuta 1.4004572400001114^http://www.w3.org/2001/XMLSchema#double
```

**Figura 25: Resultado recorrido ontología. Obtención datos del riesgo por CCAA**

Esta imagen se corresponde el primer recorrido de la ontología el cual, como hemos dicho anteriormente se corresponde con los resultados del riesgo extraído de la ontología. Como era de esperar ninguna de las comunidades tiene el mismo nivel de riesgo ya que, al depender éste de varios parámetros descritos en el apartado 4.1, el razonamiento sobre el nivel de amenaza se verá modificado.

Para comprobar que estos resultados se corresponden con los mismos que se encuentran la ontología, hemos cogido dos ejemplos, uno correspondiente a la Ciudad Autónoma de Ceuta y otro correspondiente a la Comunidad Autónoma de Extremadura y ambos resultados, en concreto el valor correspondiente al *RiesgoIntermedio*, se corresponden con los valores publicados en la Figura 25:

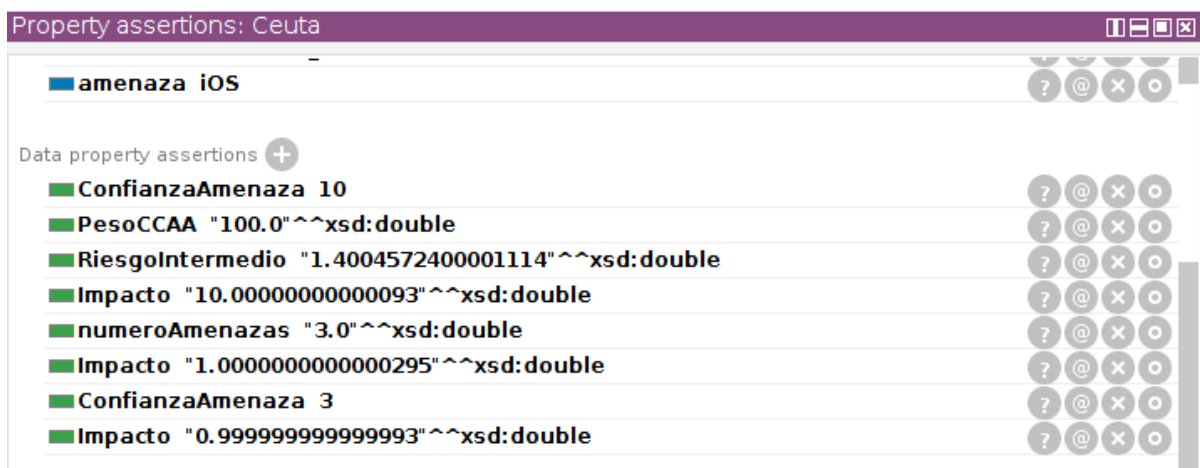


Figura 26: Resultado ontológico del riesgo de Ceuta

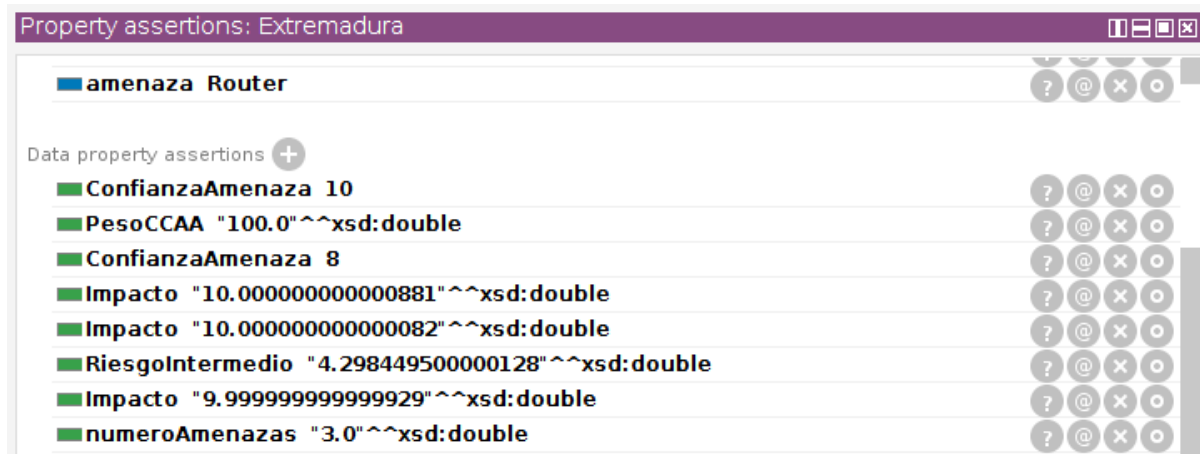


Figura 27: Resultado ontológico del riesgo de Extremadura

Como hemos dicho antes el nivel de riesgo varia de una comunidad a otra y en estas dos imágenes se puede observar muy bien esa variación debido a que los parámetros que se introducen sobre una amenaza para razonar su nivel de riesgo es distinto en cada comunidad.

Al haber 19 resultados diferentes en cuanto a resultados de activos, debido a que hay un resultado por comunidad, la siguiente visualización se corresponde con el resultado de los activos de una Comunidad Autónoma, más concretamente con la comunidad de Castilla y León:

```
Resultado_Castilla_y_León_Empresa_Autónomo_Estándar 3.225600000000087^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_Infraestructura_Critica 2.150400000000058^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Ciudadano_Avanzado 3.887146666666772^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_PYME_Estándar 3.121066666666751^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_Grande_Estándar 2.221333333333394^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_Autónomo_Tecnológico 3.347904000000091^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_Industrial 2.4752000000000667^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Ciudadano_Medio 4.3062506666667835^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Investigación_y_Universidad 2.9829333333334143^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_Grande_Tecnológica 2.795520000000076^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Empresa_PYME_Tecnológica 3.2961600000000897^http://www.w3.org/2001/XMLSchema#double
Resultado_Castilla_y_León_Ciudadano_Básico 5.166933333333474^http://www.w3.org/2001/XMLSchema#double
```

Figura 28: Resultado activos de Castilla y León

Aparecen todos los activos correspondientes a la comunidad de Castilla y León con sus valores de riesgo de sufrir una amenaza. Estos valores se emplearán posteriormente en la parte de visualización de resultados en la cual, como hemos explicado anteriormente, se realizará una tabla que contendrá, comunidad por comunidad, los resultados de los activos correspondientes.

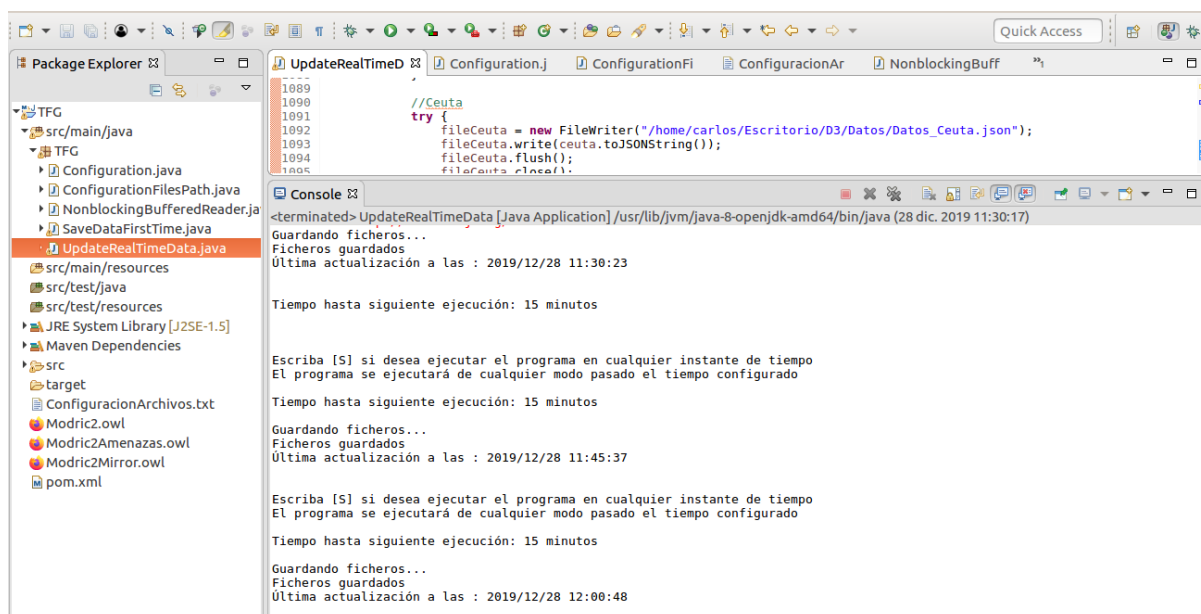
Cabe destacar que las demás comunidades autónomas representan resultados muy similares a éste, obviamente, la única diferencia radica en el valor del riesgo por activo.

## 5.2.RESULTADOS DE LOS DATOS GUARDADOS

Los siguientes resultados a analizar se corresponden con el guardado de datos. Como hemos explicado en la parte de desarrollo, el guardado de datos se realiza en tiempo real, es decir, una vez que comienza la ejecución del programa, éste no se detiene en ningún momento permitiendo el guardado o actualización de los datos cada cierto tiempo. Este tiempo se corresponde con el tiempo de lanzamiento previamente establecido en la parte de configuración.

Para realizar la simulación y al igual que se ve en la Figura 11 (tiempo de lanzamiento en segundos), este tiempo será de 15 minutos, es decir, cada vez que pase ese tiempo se producirá el guardado automático de los datos. En la imagen también se muestra varios cuadros de diálogo los nos indican lo siguiente:

- Que los ficheros se están guardando.
- Que los ficheros se han guardado correctamente.
- Tiempo que hay hasta la siguiente ejecución.
- La posibilidad de cortar la ejecución de tiempo de espera o tiempo de lanzamiento pulsando la tecla *S* en la consola del programa. Esto saltará directamente al guardado o actualización de los datos.



**Figura 29: Resultados de la ejecución del programa y del guardado de datos**

Como se puede observar en la imagen se realizaron 3 actualizaciones de datos todas en *near-real time*, sin parar la ejecución en ningún momento.

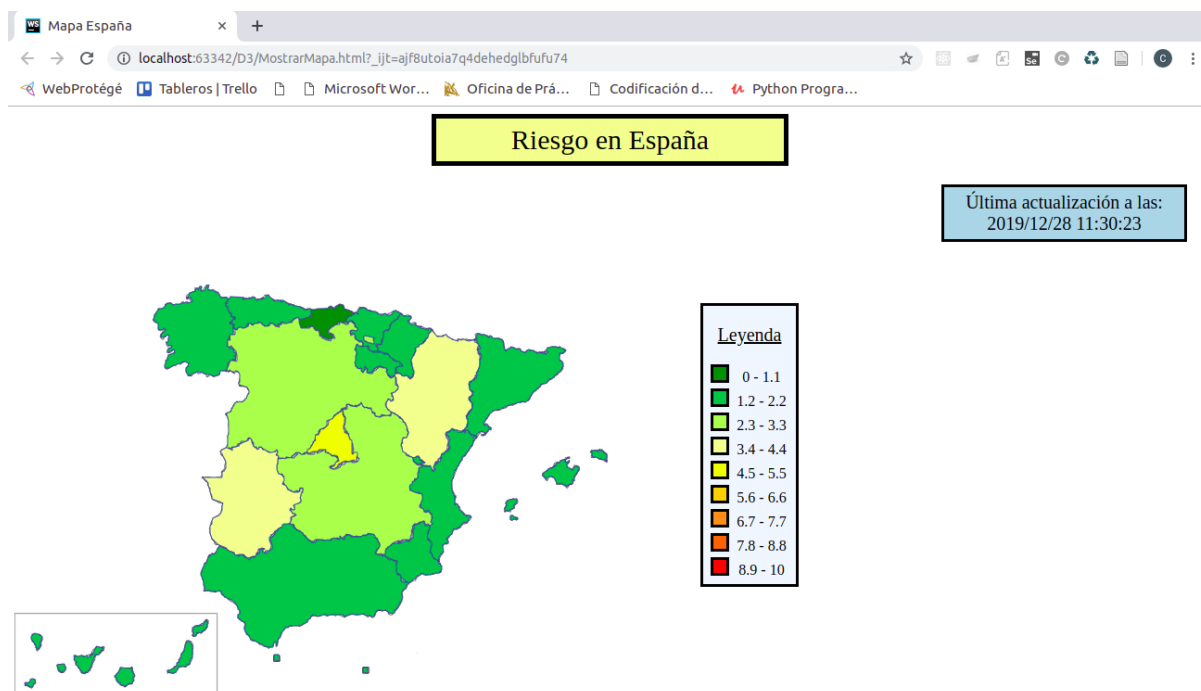
Mediante el siguiente punto veremos cómo puede variar el riesgo de sufrir un ataque solo en un intervalo de 15 minutos.

### 5.3.RESULTADOS DE LA VISUALIZACIÓN

Son los resultados más importantes del proyecto, ya que el objetivo principal del mismo, como indica el propio nombre del trabajo, hemos implementado y desarrollado un sistema de visualización en tiempo real de los riesgos que existen las diferentes Comunidades Autónomas que hay en el territorio nacional. De esta manera permitirá a los usuarios observar si sus equipos se encuentran en riesgo bajo, medio o alto de sufrir ataques cibernéticos.

La manera en la que se ha desarrollado esta simulación ha sido mediante la simulación de ataques a través de un programa ya desarrollado (5) (6) mediante el uso de *logs* a través del sistema Syslog el cual almacena las amenazas entrantes. Se emplearon tres tipos de amenazas con distinto niveles de riesgo cada una y así ver una simulación frente ataques de riesgo bajo, ataques de riesgo medio y ataques de riesgo alto obtenido los siguientes resultados:

#### 5.3.1. RESULTADOS DE RIESGO BAJO



**Figura 30: Resultado mapa de España con riesgo bajo**

Como se puede observar en la imagen, la mayoría de las comunidades están expuesta a un riesgo bajo de sufrir un ciberataque ya que los colores que predominan son los colores verdes. En comunidades como Madrid, Aragón o Extremadura existe un riesgo medio. Esto es posible debido a que no todas las comunidades tienen el mismo razonamiento de riesgo ya que este influye en varios parámetros los cuales son distintos para cada comunidad.

En esta prueba se van a analizar los activos sobre la Comunidad Autónoma de Castilla y León obteniendo así la tabla de visualización de dicha comunidad una vez que pinchamos sobre ella en el mapa anteriormente representado:





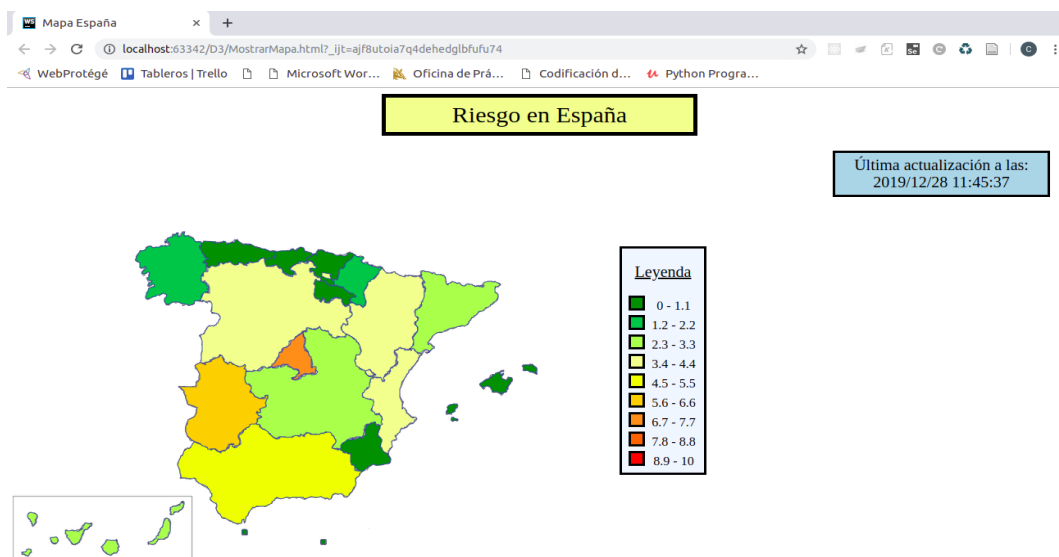
**Figura 31: Resultado activos de Castilla y León con riesgo bajo**

Hay que destacar solo en esta primera prueba los resultados coinciden con los obtenidos en la Figura 28 debido a que fueron los primeros resultados que se obtuvieron.

Como se puede observar en la imagen el riesgo que tiene la comunidad de Castilla y León es, dentro del nivel bajo, se encuentra en el subnivel más alto posible por lo que la amenaza a sufrir un ciberataque es prácticamente moderada.

Dentro de los activos existen también varios valores de niveles de riesgo de riesgo, esto indica que hay individuos más propensos a recibir un ataque como por ejemplo el *Ciudadano Básico* con un nivel de riesgo medio, que otros activos como por ejemplo la *Empresa de Infraestructura Crítica* la cual si nivel de riesgo esta incluso por debajo del nivel que hay en toda la comunidad.

### 5.3.2. RESULTADOS DE RIESGO MEDIO



**Figura 32: Resultado mapa de España con riesgo medio**

Al observar la imagen, lo primero que se resalta es que el nivel de riesgo ha subido prácticamente en todo el territorio nacional. Esto es debido a que la amenaza de los ataques simulados es mayor. En comunidades como la Comunidad de Madrid el riesgo que alcance es alto lo que hace indicar que la probabilidad de sufrir una amenaza es muchísimo mayor que otras Comunidades Autónomas como Asturias, Cantabria, País Vasco, La Rioja y Murcia entre otras.

Un dato curioso es que a pesar de que la simulación de los *logs* que contienen las amenazas es de mayor gravedad, hay comunidades algunas de las cuales anteriormente citadas (Asturias, País Vaso, La Rioja y Murcia) en los que el nivel de riesgo ha bajado. Esto es puede ser debido a múltiples parámetros de la configuración inicial para el razonamiento semántico, ya que como hemos explicado, estos parámetros son distintos en cada comunidad por lo que afectará de manera distinta a cada una de ellas.

Por otro lado, para ver también la evolución del nivel de riesgo que hay un Comunidad Autónoma en concreto, se va a representar la tabla de los activos correspondiente a Castilla y León:

Resultado	Riesgo
Castilla y León	4.298449500000128
Empresa Infraestructura Crítica	3.347904000000091
Empresa Industrial	4.9829333333334143
Empresa Autónomo Tecnológico	3.150400000000058
Empresa Autónomo Estándar	4.225600000000087
Empresa PYME Estándar	4.4752000000000667
Empresa PYME Tecnológica	2.9669333333333474
Empresa Grande Tecnológica	3.2213333333333394
Empresa Grande Estándar	2.887146666666772
Ciudadano Básico	4.121066666666751
Ciudadano Medio	3.795520000000076
Ciudadano Avanzado	4.3062506666667835
Investigación y Universidad	4.166933333333474

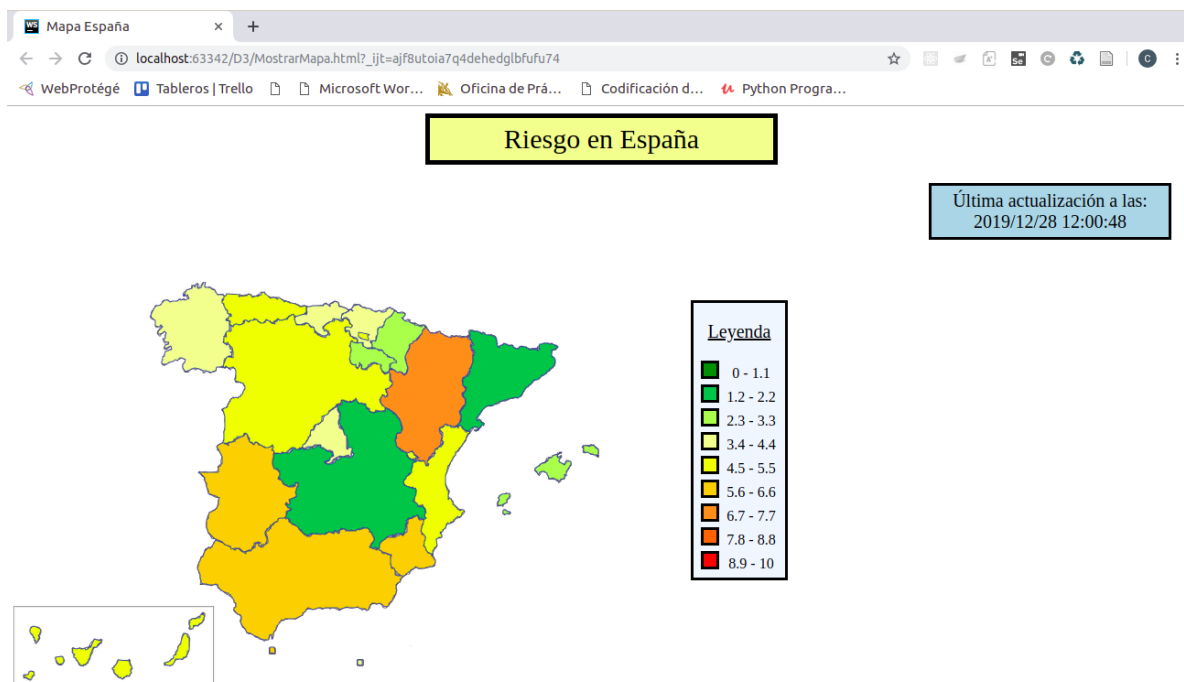
**Figura 33: Resultado activos de Castilla y León con riesgo medio**

Al comparar esta imagen con la Figura 31 se puede ver que el nivel de riesgo ha subido un poco más de un punto lo que significa que esta comunidad todavía se encuentra en un nivel moderado a sufrir un ataque. También se puede destacar que dentro del nivel medio se encuentra en el subnivel bajo muy similar al resultado realizado 15 minutos antes.

Por otro lado, si comparamos los activos estos sí que se han visto modificados. En esta segunda simulación, el individuo que se encuentra más propenso a ser víctima de un ataque es la *Empresa Industrial* con un nivel de riesgo de 4.98, un poco más alto del nivel de riesgo que hay en la comunidad. En el polo opuesto se encuentra la *Empresa Grande Estándar* con un nivel de riesgo de 2.88 puntos muy por debajo del nivel de riesgo de la comunidad.



### 5.3.3. RESULTADOS DE RIESGO ALTO



**Figura 34: Resultado mapa de España con riesgo alto**

Finalmente, esta imagen corresponde a una simulación cuyas amenazas representan un nivel alto de riesgo. Dentro de este nivel alto, nos encontramos prácticamente en el subnivel más bajo que hay por lo que el riesgo de sufrir una amenaza es bastante probable.

Comparando los resultados obtenidos con los resultados obtenidos 15 minutos atrás, se puede deducir que el nivel de riesgo ha disminuido en la Comunidad de Madrid, sin embargo, se ha visto incrementado considerablemente en otras comunidades como Aragón la cual alcanza un nivel de riesgo alto o en otras comunidades como Asturias, Cantabria o País Vasco las cuales tienen un nivel de riesgo, pero en comparación con los resultados al inicio de la simulación estos han subido drásticamente.

Otras comunidades como Castilla la Mancha o Cataluña el nivel de riesgo se mantienen prácticamente igual, con pequeñas variaciones, pero siempre manteniendo un nivel bajo.

Por último, vemos los resultados de los activos pertenecientes a la comunidad de Castilla y León para comparar su resultado con los dos anteriores. A simple vista, viendo el color con el que se encuentra coloreado esta comunidad podemos decir que su nivel de riesgo ha crecido un poco sin embargo sigue siendo moderado:

CASTILLA Y LEÓN	
Resultado	Riesgo
Castilla y León	4.895350200000392
Empresa Infraestructura Crítica	4.795520000000076
Empresa Industrial	5.121066666666751
Empresa Autónomo Tecnológico	5.347904000000091
Empresa Autónomo Estándar	5.3062506666667835
Empresa PYME Estándar	4.9829333333334143
Empresa PYME Tecnológica	4.225600000000087
Empresa Grande Tecnológica	3.9829333333334143
Empresa Grande Estándar	4.4752000000000667
Ciudadano Básico	5.2213333333333394
Ciudadano Medio	4.887146666666772
Ciudadano Avanzado	4.9669333333333474
Investigación y Universidad	5.053050000000407

**Figura 35: Resultado activos de Castilla y León con riesgo alto**

Efectivamente, el nivel de riesgo de riesgo ha crecido un poco, lo suficiente para pasar al siguiente subnivel dentro del nivel medio, pero no lo suficiente para pasar alcanzar el nivel más alto. Es nos hace indicar que todavía en dicha comunidad hay que estar más alerta debido a que la probabilidad de ser víctimas de una amenaza ha aumentado. Esto no quiere decir que los equipos de dicha comunidad estén poco protegidos y hay que protegerlos más, lo que nos indica este resultado es que es más probable que se realice un ataque sobre esta comunidad que sobre otras con niveles de riesgo más bajos.

En cuanto al análisis de los activos podemos observar que estos también han variado. En esta simulación el individuo que resulta más propenso a recibir un ataque cibernético es la *Empresa Autónoma Tecnológico* con un nivel de 5.34 puntos. En el lado opuesto se encuentra la *Empresa Grande Estándar* con un nivel de riesgo de 3.98. Ambas empresas se encuentran por encima y por debajo del nivel de la comunidad, respectivamente.

## 6. CONCLUSIONES Y LÍNEAS FUTURAS

### 6.1. CONCLUSIONES

Como conclusiones hay que establecer que el programa desarrollado cumple con los requisitos funcionales previamente propuestos ya que permite un guardado o actualización de resultados en tiempo real lo que garantiza que la posterior visualización de estos también lo sea cumplirse así con los objetivos marcados.

Esta visualización permite ver los niveles de riesgo que hay en las Comunidades Autónomas en España, sin embargo, esto no garantiza que los equipos de una comunidad no puedan sufrir un ataque cibernético de tipo malware a pesar de que el riesgo en esa comunidad sea bajo ya que existen otros tipos de ataques que pueden dañar los dispositivos de los usuarios debido a que se pueden producir por otros medios ya sea por ingeniería social, suplantación de identidad o ataques de denegación de servicio. Además, la visualización no muestra el tipo de ataques pueden sufrir los equipos.

Hay que tener en cuenta que un riesgo bajo o casi nulo no significa que una comunidad o una infraestructura no pueda ser víctima de un ataque. Por el caso contrario una comunidad o una infraestructura con un nivel de riesgo alto puede no ser víctima de un ataque o los daños producidos por el mismo pueden ser casi nulos. Lo que en realidad nos muestra la simulación es la probabilidad de que la comunidad o sus activos se conviertan en víctimas de cualquier tipo de ataques, por lo tanto, a niveles bajos esta probabilidad disminuye y por el contrario a niveles altos, la probabilidad aumente.

Cabe destacar que es necesario siempre tener todos los equipos bien protegidos y estar constantemente actualizados ya que si no se hiciera el riesgo de sufrir un ataque aumentaría poniendo tanto en peligro al equipo personal como en el ámbito empresarial, poner en riesgo a toda la entidad pudiendo producir una exposición de información no deseada. También es muy aconsejable tener un seguro frente a daños cibernéticos con el objetivo final de reducir lo máximo posible los daños producidos por el ataque.

Uno de los aspectos negativos que se han encontrado en el desarrollo visual del trabajo es utilización del lenguaje HTML como solución del sistema de visualización debido a que no se ha podido encontrar ningún otro programa que permita poder dibujar y poder representar las diferentes Comunidades Autónomas que hay en España. Esto ha limitado en varios aspectos el desarrollo del trabajo, perdiendo así el poder realizar una visualización más iterativa con el usuario.

En cuanto al programa realizado en Eclipse para la obtención de resultados y su posterior guardado es el elemento fundamental del trabajo debido a que nos permite realizar esa visualización en tiempo real gracias a la actualización de datos que vamos a representar. Además, nos existen varias posibilidades, en cuanto al tiempo de almacenamiento se refiere, ya que se puede establecer un tiempo de configuración de acuerdo con las necesidades del usuario o también se puede guardar cuando el usuario lo desee acortando y reiniciando ese tiempo anteriormente configurad. Además, a pesar de que el programa este continuamente en funcionamiento, ese tiempo de almacenamiento puede ser modificado sin afectar al transcurso de este.

Finalmente, hay que mencionar que se han añadido unas pequeñas mejoras visuales para conocer cuando se ha producido la última actualización de los datos, lo que facilita al usuario realizar un mejor seguimiento de mismos ya que puede comprobar que efectivamente cada cierto tiempo los valores del riesgo se modifican, no solo mirando el mapa de España sino también con esta fecha de actualización.

También me gustaría concluir con este trabajo diciendo que me ha servido de gran ayuda para familiarizarme con nuevas tecnologías como son las ontologías y como a partir de una serie de valores, realizando unos razonamientos semánticos, se puede obtener información muy valiosa

acerca de cómo un mismo ataque cibernético o ataque físico puede influir de distintas maneras en comunidades diferentes dependiendo de una serie de factores que son característicos de cada comunidad.

## 6.2.LÍNEAS FUTURAS

En cuanto a líneas futuras, se puede plantear varias ideas que mejoren el trabajo ya realizado. En concreto se plantean dos mejoras, una en cuanto a la obtención de datos y otra en cuanto a la visualización de éstos.

En primer lugar, una posible mejora que se puede realizar es que para tener más detalle acerca del riesgo, que éste no solo se obtenga a nivel autonómico sino también provincial, es decir, que se produzca un razonamiento semántico de amenazas también de cada provincia que hay en España.

En segundo lugar, también se puede realizar mejoras en la visualización haciendo que ésta sea más iterativa. De esta forma se podría visualizar el riesgo que hay en todo el territorio nacional. A continuación, si queremos obtener más información sobre el nivel de riesgo a nivel autonómico, se pincharía sobre el mapa y aparecerían las comunidades tal y como las hemos representado en este trabajo. Finalmente podrías acceder a un tercer nivel correspondiente con el riesgo a nivel provincial. Así, mediante este mecanismo, podríamos obtener más información del nivel de riesgo calculado.

Por otra parte, también se podría realizar una serie de mejoras en cuanto a la información presentada al usuario permitiendo a este saber si el tipo de amenazas entrantes, es decir, si son troyanos, gusanos, virus informáticos... Además, se podría proporcionar información de cómo prevenir este tipo de amenazas lo máximo posible.

## 7. BIBLIOGRAFÍA

- [1] **Cibernos Comunicación.** Cibernos.com. *Cibernos*. 1 de octubre de 2019. [En línea].  
Available: <https://www.cibernos.com/blog/cual-es-el-mayor-riesgo-de-ciberseguridad-para-mi-empresa>.
- [2] **Silva, José.** Seguridad informática: amenazas comunes. 14 de enero de 2019. [En línea].  
Available: <http://www.josilva.com/blog/Posts/show/seguridad-informatica-amenazas-comunes-888>.
- [3] **Conexionesan.** 14 de Septiembre de 2018. [En línea]. Available:  
<https://www.esan.edu.pe/apuntes-empresariales/2018/09/como-gestionar-los-riesgos-de-ciberseguridad/>.
- [4] **Amutio Gómez, Miguel Ángel, Candau, Javier y Mañas, José Antonio.** *MAGERIT- versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*. Madrid : Ministerio de Hacienda y Administraciones Públicas. 2012. [En línea]. Available:  
<https://www.ccn-cert.cni.es/documentos-publicos/1789-magerit-libro-i-metodo/file.html>
- [5] **Vega-Barbas, M; Villagrà, V. A.; Monje, F; Riesco, R; Larriva-Novo, X; Berrocal, J.** *Ontology-Based System for Dynamic Risk Management in Administrative Domains*. Universidad Politécnica de Madrid, España. Appl. Sci. 2019,9,4547. [En línea]. Available:  
<https://www.mdpi.com/2076-3417/9/21/4547>
- [6] **Monje Real, Fernando.** *Design and development of a translation and enforcement module for cybersecurity policies*. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S.I.T. Telecomunicaciones (UPM), Madrid. 2018. [En línea]. Available: <http://oa.upm.es/51999/>
- [7] **McGuinness, Deborah y Van Harmelen, Frank.** Lenguajes de Ontologías Web (OWL). 10 de febrero de 2014. [En línea]. Available: <https://www.w3.org/2007/09/OWL-Overview-es.html>.
- [8] **Horridge, Matthew.** *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. The University of Manchester, England. 2011.
- [9] **Boley, Harold, Tabet, Said y Wagner, Gerd.** *Desing Rationale of RuleML: A Markup Language for Semantic Web Rules*. Faculty of Technology Management Eindhoven University of Technology, Netherlands.
- [10] **Horrocks, Ian; Pathel-Schneider, Peter F.; Boley, Harold; Tabet, Said; Grosz, Benjamin; Dean, Mike.** SWRL: A Semantic Web Rule Language combining OWL and RuleML. 21 de mayo de 2004. [En línea]. Available:  
<https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- [11] **Kemayo.** Maphilight: Image map mouseover highlighting. 5 de marzo de 2008. [En línea]. Available: <https://davidlynch.org/blog/2008/03/maphilight-image-map-mouseover-highlighting/>

- [12] **Protégé, Wikipedia, la enciclopedia libre.** [En línea].Available:  
[https://es.wikipedia.org/wiki/Prot%C3%A9g%C3%A9\\_\(software\)](https://es.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(software)).
- [13] **Apache Jena, Jena Ontology API. 2019.** [En línea].Available:  
<https://jena.apache.org/documentation/ontology/>.
- [14] **WebStorm.**2019. [En línea].Available: <https://www.jetbrains.com/es-es/webstorm/>.
- [15] **Herrera, Javier Flores.** Qué es HTML. 25 de agsto de 2015. [En línea]. Available:  
<https://codigofacilito.com/articulos/que-es-html>.
- [16] **Gracia, Luis Miguel.** Qué es Apache Jena. 27 de julio de 2012. [En línea]. Available:  
<https://unpocodejava.com/2012/07/27/que-es-apache-jena/>.
- [17] **HashMap, HashMap en Java.** [En línea]. Available: <https://guru99.es/working-with-hashmaps/>.
- [18] **Interface JsonObject.** 2015. [En línea]. Available:  
<https://docs.oracle.com/javaee/7/api/javax/json/JsonObject.html>.
- [19] **Interface JsonArray.** 2015. [En línea]. Available:  
<https://docs.oracle.com/javaee/7/api/javax/json/JsonArray.html>.
- [20] **Class FileWriter.** 2018. [En línea]. Available:  
<https://docs.oracle.com/javase/7/docs/api/java/io/FileWriter.html>.
- [21] **Gómez, David Raygoza.** Convertir objetos Java a JSON y de regreso. 10 de junio de 2018.  
[En línea]. Available: <https://medium.com/el-acordeon-del-programador/convertir-objetos-java-a-json-y-de-regreso-1077500d78f7>.
- [22] **MDN Web Docs: JSON.parse().** 2019. [En línea]. Available:  
[https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\\_globales/JSON/parse](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/JSON/parse).
- [23] **MDN Web Docs: JSON.stringify().** 2019. [En línea]. Available:  
[https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\\_globales/JSON/stringify](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/JSON/stringify).

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

### A.1 INTRODUCCIÓN

El sistema que hemos desarrollado puede llevar consigo una serie de impactos sociales, económicos, éticos y ambientales. El objetivo de este proyecto es su uso, principalmente, en empresas, aunque también puede ser usado por particulares, con el fin de poder prevenir ataques cibernéticos ya sean en los equipos informáticos pertenecientes las propias empresas como dispositivos personales como puede ser un teléfono móvil. Este sistema desarrollado no garantiza que dichos dispositivos se encuentren protegidos, por lo que un descuido del propio usuario puede garantizar impactos negativos.

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

#### **Impacto ético**

De forma indirecta, el principal impacto ético que puede haber es el de ser víctima de un ataque mediante el cual se produzca un robo de datos ya sean datos personales si el ataque se ha producido a un dispositivo personal, o la usurpación de una gran cantidad de datos protegidos si el ataque se realiza a una empresa. Por lo tanto, el programa desarrollado resulta útil para conocer el nivel de riesgo que hay, medirlo, evaluarlo y realizar acciones que disminuyan todo lo posible los daños.

Directamente, el sistema no supone ningún impacto ético ya que simplemente los datos que se analizan corresponden a las amenazas que pueden atacar nuestros sistemas. Por lo tanto, no estamos trabajando con datos personales que puedan suponer una amenaza contra la ley de protección de datos.

#### **Impacto social**

Una de las finalidades de este proyecto es garantizar y concienciar a los usuarios y a las empresas del nivel de riesgo a los que se encuentran expuesto sus dispositivos electrónicos si estos no se protegen adecuadamente ya que un ataque les puede robar información importante y no solo eso, sino que también esto puede suponer un problema de reputación o de imagen en la empresa que ha sido atacada. Por lo tanto, con lo que se pretende con este sistema es evaluar un nivel de riesgo y así poder mitigar en la medida de lo que se pueda los daños que se pueden causar al ser víctimas de un ataque cibernético.

#### **Impacto medioambiental**

Desde un primer punto de vista, parece ser que no existe ningún impacto medioambiental grave. Sin embargo, si analizamos la situación de manera más detenida, el programa realizado se tiene que ejecutar en uno o varios ordenadores los cuales consumen bastante energía. Por otra parte, la visualización de los resultados se tiene que representar sobre una pantalla, ya sea la del propio ordenador, un monitor o cualquier dispositivo que permita una visualización. Al ser en tiempo real, estos dispositivos se encontrarán constantemente en funcionamiento por lo que el consumo de energía por parte de los mismo también supondrá un alto impacto ambiental.

De forma indirecta, los dispositivos empleados anteriormente, llegará un momento en el cual ya no sirvan y tendrán que ser reemplazados por otros más nuevos y mejores. Por lo tanto, dichos dispositivos, al ser no renovables, tendrán que ser eliminados en un punto limpio lo que también provoca impactos medioambientales.



En este tipo de impacto, de forma directa, encontramos nuestro propio equipo de trabajo, más concretamente un ordenador personal el cual consume energía y dispone de una vida útil.

### **Impacto económico**

Este impacto afecta de manera indirecta a nuestro proyecto debido a que, gracias a nuestro programa, las empresas o usuarios tendrán una visión del nivel de riesgo que tienen de ser víctimas de una amenaza. Con el objetivo de disminuir los daños producidos por un ataque, es necesario proteger los equipos que pueden verse afectados, ya sea mediante seguros u otro tipo de protección lo que supondrá al usuario o a la empresa gastarse una cantidad de dinero para la protección.

## **A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS**

El impacto principal es un conjunto de todos los impactos anteriores ya que no se puede destacar uno por encima.

Esto es debido a que, en primer lugar, para conocer el nivel de riesgo, es necesario ejecutar el programa y mantenerlo en ejecución todo el tiempo para que se produzca una actualización de los datos en tiempo real, el ordenador que está ejecutando dicho programa está consumiendo energía tanto para evitar que la batería del dispositivo se agote como para la representación visual sobre una pantalla o monitor (impacto medioambiental).

Por otro lado, tenemos una empresa la cual tiene un riesgo, ya sea alto, bajo o medio, de ser víctima de un ciberataque. Dicha empresa al cabo de un tiempo recibe un malware por medio de un troyano el cual afecta a todos los dispositivos electrónicos de la empresa robando así información personal sobre los clientes asociados a la empresa (impacto ético y social).

Por suerte, el dueño de la empresa tiene contratado un seguro de ciberriesgo el cual cubre un alto porcentaje de los daños producidos. Por lo tanto, para contratar este seguro se tiene que invertir una gran cantidad de dinero (impacto económico). En el hipotético caso en el que la empresa no tenga contratado ningún seguro, dicha empresa tendría que recuperar todos los datos que han sido substraídos lo que conllevaría una inversión de dinero además de que la reputación y la imagen de la empresa se vería perjudicada (impacto económico y social).

## **A.4 CONCLUSIONES**

A pesar de que se ha desarrollado un sistema que muestra el nivel de riesgo que puede existir en las diferentes Comunidades Autónomas, éste no nos garantiza que, en el caso de que exista un nivel bajo, estemos protegidos frente a una amenaza. Por lo tanto, un exceso de confianza por parte de las empresas o de los usuarios al no proteger sus dispositivos, supondría un aumento de sufrir un ataque. Como conclusión siempre tenemos que proteger a nuestros equipos y realizar actualizaciones periódicamente con el objetivo de disminuir ser víctimas de un ataque.

## ANEXO B: PRESUPUESTO ECONÓMICO

### COSTE DE MANO DE OBRA (coste directo)

Horas	Precio/hora	Total
472	15 €	<b>7.080 €</b>

### COSTE DE RECURSOS MATERIALES (coste directo)

	Precio de compra	Uso en meses	Amortización (en años)	Total
Ordenador personal (Software incluido)	828,95 €	9	5	124,34 €

### COSTE TOTAL DE RECURSOS MATERIALES

**124,34 €**

GASTOS GENERALES (costes indirectos)	15%	sobre CD	<b>1080,75 €</b>
BENEFICIO INDUSTRIAL	6%	sobre CD+CI	<b>497,11 €</b>

### SUBTOTAL PRESUPUESTO

**8.782,20 €**

### IVA APLICABLE

21%

**1.844,26 €**

### TOTAL PRESUPUESTO

**10.625,46 €**