UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior
de
Ingeniería y Sistemas de Telecomunicación

PROYECTO FIN DE GRADO

SISTEMA DE SELECCIÓN AUTOMÁTICA DE SOLUCIONES TECNOLÓGICAS DE SEGURIDAD

ALEJANDRO RUEDA PÉREZ

GRADO EN INGENIERÍA TELEMÁTICA

Julio de 2014

TELECOMUNICACIÓN

Campus Sur

POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Sistema de Selección Automática de Soluciones Tecnológicas de Seguridad

AUTOR: Alejandro Rueda Pérez

TITULACIÓN: Telemática

TUTOR (o Director en su caso): Lourdes López Santidrián

DEPARTAMENTO: Telemática

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: Elena Blanco Martín

VOCAL: Lourdes López Santidrián

SECRETARIO: Gregorio Rubio Cifuentes

Fecha de lectura: 15/07/2014

Calificación:

El Secretario,

Julio de 2014

AGRADECIMIENTOS

Quiero empezar por dar las gracias a mi padre, mi madre y mi hermana porque sin ellos no hubiera podido conseguir llegar a este punto de mi vida.

También, por supuesto, a mi tutora Lourdes y a, como digo yo, mi segundo tutor José Antonio porque me acogieron y confiaron en mí en una situación que no todo el mundo lo hubiera hecho.

Agradecer a toda la demás gente que, sin dudarlo, me han aguantado horas y horas hablando del proyecto e incluso han aportado un poco de calidad en el código como mis amigos del barrio, amigos de la universidad y Timpurinkuja family.

Desde luego no se me olvida Ángela que, aunque ha estado lejos de mi durante este año, siempre estaba 24 horas disponible y apoyándome.

Me gustaría, por último, agradecer a todas aquellas personas que no solo me han ayudado realizar este proyecto, sino que me han hecho ser quien soy.

Más que ayer pero menos que mañana…

RESUMEN

Desde la aparición de Internet, hace ya más de 20 años ha existido por parte de diversos sectores de la sociedad, científicos, empresas, usuarios, etc. la inquietud por la aplicación de esta tecnología a lo que se ha dado en llamar “El Internet de las Cosas”, que no es más que el control a distancia de cualquier elemento útil o necesario para la vida cotidiana y la industria.

Sin embargo el desarrollo masivo de aplicaciones orientadas a esto, no ha evolucionado hasta que no se han producido avances importantes en dos campos: por un lado, en las Redes Inalámbricas de Sensores (WSN), redes compuestas por un conjunto de pequeños dispositivos capaces de transmitir la información que recogen, haciéndola llegar desde su propia red inalámbrica, a otras de amplia cobertura y por otro con la miniaturización cada vez mayor de dispositivos capaces de tener una autonomía suficiente como para procesar datos e interconectarse entre sí.

Al igual que en las redes de ordenadores convencionales, las WSN se pueden ver comprometidas en lo que a seguridad se refiere, ya que la masiva implementación de estas redes hará que millones de Terabytes de datos, muchas veces comprometidos o sometidos a estrictas Leyes de protección de los mismos, circulen en la sociedad de la información, de forma que lo que nace como una ventaja muy interesante para sus usuarios, puede convertirse en una pesadilla debido a la amenaza constante hacia los servicios mínimos de seguridad que las compañías desarrolladoras han de garantizar a los usuarios de sus aplicaciones

Éstas, y con el objetivo de proveer un ámbito de seguridad mínimo, deben de realizar un minucioso estudio de la aplicación en particular que se quiere ofrecer con una WSN y también de las características específicas de la red ya que, al estar formadas por dispositivos prácticamente diminutos, pueden tener ciertas limitaciones en cuanto al tamaño de la batería, capacidad de procesamiento, memoria, etc.

El presente proyecto desarrolla una aplicación, única, ya que en la actualidad no existe un software con similares características y que aporta un avance importante en dos campos principalmente: por un lado ayudará a los usuarios que deseen desplegar una aplicación en una red WSN a determinar de forma automática cuales son los mecanismos y servicios específicos de seguridad que se han de implementar en dicha red para esa aplicación concreta y, por otro lado proporcionará un apoyo extra a expertos de seguridad que estén investigando en la materia ya que, servirá de plataforma de pruebas para centralizar la información sobre seguridad que se tengan en ese momento en una base de conocimientos única, proporcionando también un método útil de prueba para posibles escenarios virtuales.

ABSTRACT

It has been more than 20 years since the Internet appeared and with it, scientists, companies, users, etc. have been wanted to apply this technology to their environment which means to control remotely devices, which are useful for the industry or aspects of the daily life.

However, the huge development of these applications oriented to that use, has not evolve till some important researches has been occurred in two fields: on one hand, the field of the Wireless Sensor Networks (WSN) which are networks composed of little devices that are able to transmit the information that they gather making it to pass through from their wireless network to other wider networks and on the other hand with the increase of the miniaturization of the devices which are able to work in autonomous mode so that to process data and connect to each other.

WSN could be compromised in the matter of security as well as the conventional computer networks, due to the massive implementation of this kind of networks will cause that millions of Terabytes of data will be going around in the information society, thus what it is thought at first as an interesting advantage for people, could turn to be a nightmare because of the continuous threat to the minimal security services that developing companies must guarantee their applications users.

These companies, and with the aim to provide a minimal security realm, they have to do a strict research about the application that they want to implement in one WSN and the specific characteristics of the network as they are made by tiny devices so that they could have certain limitations related to the battery, throughput, memory, etc.

This project develops a unique application since, nowadays, there is not any software with similar characteristics and it will be really helpful in mainly two areas: on one side, it will help users who want to deploy an application in one WSN to determine in an automatically way, which ones security services and mechanisms are those which is necessary to implement in that network for the concrete application and, on the other side, it will provide an extra help for the security experts who are researching in wireless sensor network security so that ti will an exceptional platform in order to centralize information about security in the Wireless Sensor Networks in an exclusive knowledge base, providing at the same time a useful method to test virtual scenarios.

INDICE DE CONTENIDOS

ACRONIMOS

WSN: Wireless Sensor Network

IoT: Internet of Things

OSI: Open System Interconnection

SCADA: Supervisory control and data acquisition

ICS: Industrial Control System

WAN: Wide Area Network

LAN: Local Area Network

RTU: Remote Telemetry Units

PLC: Programmable Logic Controller

MIT: Massachusetts Institute of Technology

RFID: Radio Frequency Identification

DoS: Deny of service

P2P: Peer-to-Peer

JSF: Java Server Faces

JEE: Java Enterprise Edition

JPA: Java Persistence API

JDBC: Java Database Conectivity

SQL: Structures Query Language

PDPS-IoT: Provisión De Políticas de Seguridad en IoT

BES: Business Expert System

LES: Legal Expert System

TSES: Technological Solutions Expert System

FDES: Final Decision Expert System

ITU: International Telecommunication Union

QoS: Quality of Service

AWARE: Accessible Wearable Device Platform for Smart Environments

CITSEM: Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad

API: Application Programming Interface

XHTML: eXtensible HyperText Markup Language

LODP: Ley Orgánica de Protección de Datos

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El desarrollo exponencial en los últimos veinte años de Internet, ha llevado consigo la aparición de muchas aplicaciones asociadas a esta tecnología.

La posibilidad de controlar a distancia cualquier objeto necesario no solo para la industria o puesto de trabajo, sino también para la vida cotidiana ha abierto un campo de posibilidades a muchas empresas, investigadores, etc. que han visto en él tanto un avance tecnológico importante como la posibilidad de un negocio interesante. Es lo que se ha dado en llamar el "Internet de las Cosas".

Así, lo que al principio sonaba lejano o difícil fue tomando forma con la aparición y desarrollo de las Redes Inalámbricas de Sensores (WSN), un conjunto de pequeños dispositivos capaces de transmitir la información que recogen, haciéndola llegar desde su propia red inalámbrica, a otras de amplia cobertura como Internet.

Con el masivo crecimiento a nivel global de la tecnología WSN, ha aparecido la necesidad de aplicar en dichas redes un cierto nivel de seguridad que permita a cualquier entidad proveedora de servicios del "Internet de las Cosas" ofrecer un mínimo grado tanto de fiabilidad como de seguridad.

Actualmente, no se dispone de un sistema automático de determinación de los servicios y mecanismos de seguridad necesarios en el "Internet de las Cosas". Existen muchos y diferentes tipos de aplicaciones para esta tecnología emergente, que teniendo en cuenta diversos factores, de los cuales los más destacados son los legales y los físicos de la red, deben considerar la implementación de un sistema de seguridad diferente para cada caso particular de aplicación.

Hasta la fecha, la solución a esta problemática se ha buscado dejando que, en el mejor de los casos a través de una consultoría encargada de realizar un informe acerca de qué soluciones se han de aplicar para cada caso concreto o, en otros casos haciendo que cada desarrollador lo decidiera con sus propios criterios, con los inconvenientes que esto puede acarrear.

En este proyecto se pretende realizar un sistema de selección automática de soluciones tecnológicas de seguridad aplicado a los productos y servicios de "Internet de las Cosas".

El avance que aporta esta solución sobre el estado del arte es resolver de forma automática las mejores soluciones de seguridad para el caso concreto del servicio que se trate, sin gastos de consultoría, homogeneizar las soluciones que se aplican a casos similares, y obtener las soluciones más avanzadas y fiables mediante la correcta gestión y alimentación de la base de conocimientos.

ESTUDIO DEL ESTADO DE LA CUESTIÓN

1 El Internet de las Cosas

1.1 Introducción

El Internet de las cosas, a priori, parece un concepto nuevo. Sin embargo, es algo que se ha ido forjando poco a poco, apoyado principalmente por el desarrollo tecnológico en los campos de Internet, capacidad de procesamiento, Web 2.0, Redes Inalámbricas de Sensores (WSN), etc. [1]

Este concepto representa la idea de que cualquier objeto que necesite para funcionar una interacción con el ser humano, desde un mero interruptor de luz hasta un electrodoméstico, estará conectado a Internet, lo que conlleva que desde cualquier lugar donde haya una conexión a Internet, si una persona autorizada quiere manejar ese dispositivo, lo podrá hacer aunque se encuentre a miles de kilómetros de distancia. [2]

Esto, por una parte ocasionará que haya millones de Terabytes de datos de información procedentes de miles de millones de dispositivos comunicándose e interactuando con uno o varios usuarios cada uno, en tiempo real, y por otra, la necesidad de implementar un importante nivel de seguridad para proteger al usuario y la información que circula por las redes, en muchas ocasiones datos sensibles, de posibles ataques de ciberpiratería.

Como ejemplo de lo anterior se podría pensar en el interruptor de luz de una vivienda. Éste ofrecería dos estados y un log con un sello de tiempo cada vez que pasa de un estado a otro. Toda esta información estaría en Internet, siendo vulnerable de poder ser pirateada por un tercero quien podría saber, por ejemplo, cuándo no hay nadie dentro de la casa del propietario, y utilizar esta información para cometer robos, etc. [1]

1.2 Antecedentes

La primera vez que se habló del concepto de “Internet of Things” fue cuando el Massachusetts Institute of Technology (MIT), en 1999, empezó a desarrollar una red que comunicaría los objetos físicos a través de la tecnología Radio Frequency Identification (RDIF).

Su objetivo principal era que los ordenadores fueran capaces de entender el mundo real, como dijo su cofundador Kevin Ashton en 2002 en la revista Forbes: “We need an internet for things, a standardized way for computers to understand the real world” [3] siendo posteriormente titulado como “Internet of Things”.

Neil Gershenfeld, colaborador en el departamento de Media del MIT, también predijo en su libro “When Things Start to Think” [4] que los objetos tendrían un

papel importante en Internet manifestando: "in retrospect it looks like the rapid growth of the World Wide Web may have been just the trigger charge that is now setting off the real explosion, as things start to use the Net".

A partir de entonces se usó dicho término para denominar ciertos libros y conferencias [5] [6] pero no sería hasta 2008 cuando tuvo lugar la primera conferencia acerca de este tema [7]. En principio, el término estaba muy relacionado con RFID pero finalmente se consolidó cuando se utilizó oficialmente en las conferencias de RFID de 2006: "From RFID to the Internet of Things" [8] y de 2007: "RFID: Towards the Internet of Things" [9].

Finalmente, la Comisión Europea acabó aceptando en 2009 que el "Internet de las Cosas" era el siguiente paso en la evolución de Internet. [10]

En la actualidad, los estudios realizados por algunas Compañías del sector como Cisco prevé que en 2020 habrá cerca de 50 billones de dispositivos de este tipo conectados entre ellos y todo esto dentro de un mercado que excederá los 14 trillones de dólares porque de acuerdo con la compañía, el Internet of Things (IoT) es una tecnología que para nada es futurista, sino que, ya es realidad y ya está aquí. [11]

1.3 Redes Inalámbricas de Sensores (WSN)

Una Red Inalámbrica de Sensores es una red formada por dispositivos generalmente diminutos que están equipados con sensores y que recogen información de forma continua del ambiente con el objetivo de transmitirla. Entre ellos se pueden formar redes ad-hoc que son las más comunes y que, gracias a su capacidad de comunicación inalámbrica, le permiten no tener una infraestructura física fija ni tampoco administración central.

El concepto ad-hoc es el principio en el que se basan este tipo de redes ya que el término se refiere a cualquier red en la que no hay un nodo central, sino que éstos trabajan en conjunto para poder encaminar los paquetes desde su fuente a su destino.

Son enormes las posibilidades que ofrecen este tipo de redes donde se pueden desplegar multitud de dispositivos diminutos y que además, no necesitan de una estructura preestablecida. Algunos campos en los que se podrían utilizar debido a la flexibilidad que ofrecen con respecto a su arquitectura, son por ejemplo: industrial, domótica, ambiental, militar… [12]

1.3.1 Arquitectura y componentes

El objetivo típico de las redes WSN es el de realizar mediciones sobre el entorno en el que están desplegadas, transformar la información a digital y transmitirla al exterior de la red a través de un elemento llamado Gateway. Una vez que la información está en otra red, como por ejemplo podría ser Internet, el objetivo es llegar a una estación base donde será almacenada y analizada.

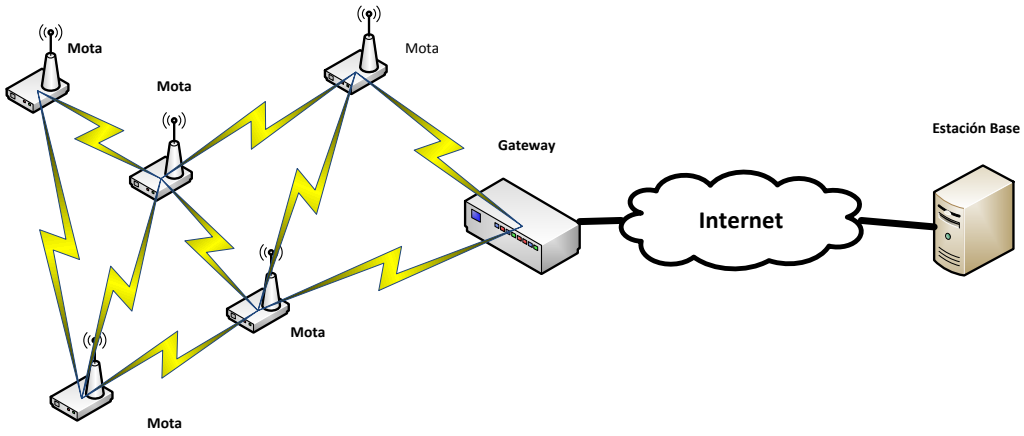


Figura 1. Arquitectura general de una WSN

Por tanto, como se observa en la Figura 1, se distinguen los siguientes componentes que intervienen en este tipo de redes: [12]

- *Nodos inalámbricos:* Son los también llamados "motas" debido a su tamaño y ligereza. Su función dentro del entramado de la red es captar información del entorno en el que se encuentran[1] y transmitirla hacia el destinatario que, como se ha dicho antes, en un caso general sería a su Gateway asociado.

  Los nodos de una red inalámbrica de sensores suelen ser de tamaño pequeño ya que esto es lo que da utilizad a este tipo de redes y por tanto tienen ciertas limitaciones en su arquitectura interna como puede ser en: alimentación, memoria, procesamiento, comunicación inalámbrica, etc. Esto último, como se verá posteriormente, es lo que puede provocar en muchos casos vulnerabilidades de seguridad en la red.

- *Gateway o puerta de enlace:* Es un elemento que no tiene sensores sino que su única función es conectar dos redes: por un lado la red de sensores encargada de recoger la información de las motas para poder enviarla a la estación base y por otro la red a la que tiene acceso la estación base.
  A su vez la comunicación puede ser unidireccional, es decir, que únicamente la estación base reciba la información de las motas de la red inalámbrica de sensores o bidireccional, que también desde la estación base se mande algún comando de control hacia la WSN.
  En el caso de la Figura 1, la comunicación es bidireccional y el Gateway comunica la red inalámbrica de sensores con Internet que es donde está conectada la estación base.
- *Estaciones base:* Se puede tratar tanto de un servidor como de un simple ordenador común. Actúa como el destinatario final de la información

[1] El tipo de información que se esté captando depende del tipo de sensores con el que esa mota esté equipada. No necesariamente tiene que ser un solo sensor.

recopilada por la red WSN. Como se ha mencionado anteriormente, los usuarios también pueden, en algunos casos, acceder e incluso modificar aspectos de control de la red.

1.3.2 Topología

Una vez conocidos los elementos que pueden formar parte de una Red Inalámbrica de Sensores, es preciso especificar los tipos de topología en los que se pueden estructurar. [12]

Topología en estrella: Los nodos no intercambian información entre ellos porque todos están conectados directamente con la puerta de enlace como se observa en la Figura 2. Este tipo es el que menor gasto de energía conlleva pero por el contrario, tiene limitaciones en cuanto a la distancia entre el nodo y la puerta de enlace ya que no se producen retransmisiones intermedias.

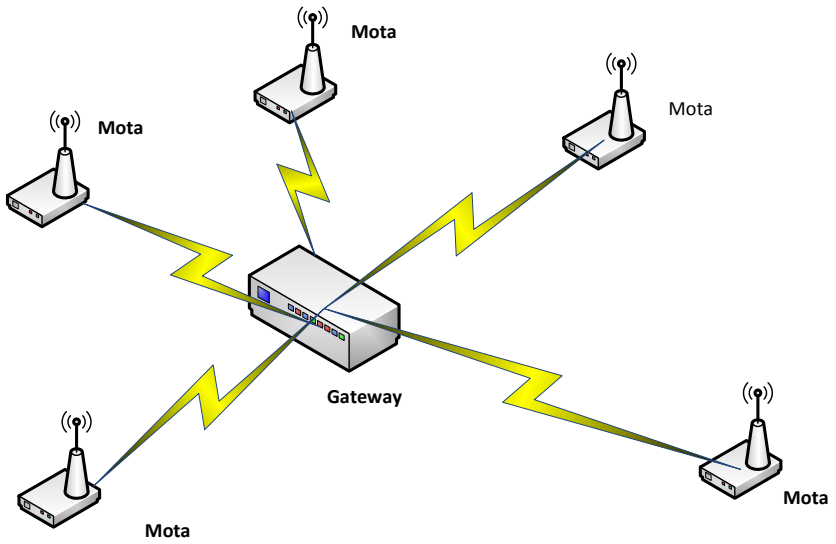


Figura 2. Topología en estrella en las WSN

- *Topología en malla:* En la topología mallada, los nodos actúan también como "routers" y así, pueden buscar caminos alternativos a través de algún tipo de protocolo de enrutamiento. La ventaja de este tipo es que, en teoría, la extensión de redes con esta topología sería ilimitada y se podría establecer controles de fallos. Sin embargo, como desventajas estarían entre otras los retrasos de los paquetes.

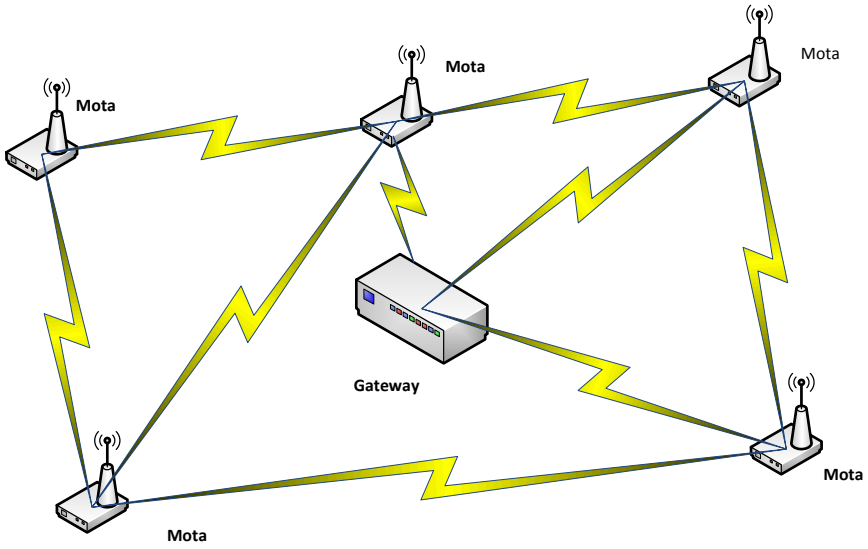


Figura 3. Topología mallada en las WSN

- *Topología híbrida estrella-malla:* Como su propio nombre indica, este tipo es una mezcla de las dos topologías cuyo objetivo es combinar las ventajas de cada una de ellas. Intenta dar la posibilidad de enrutamiento en la red a la vez que ahorra energía utilizando nodos finales.

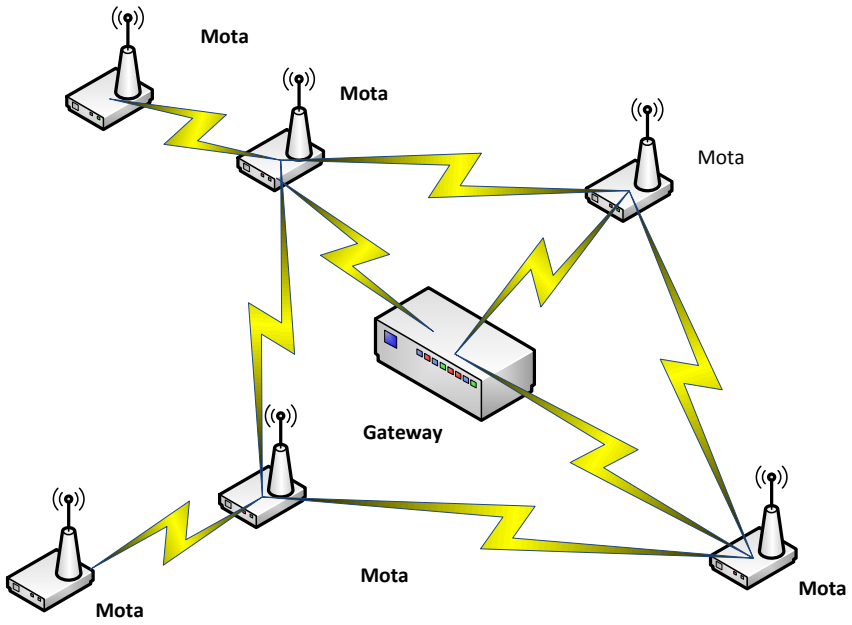


Figura 4. Topología híbrida en las WSN

1.3.3 Aplicaciones en las Redes Inalámbricas de Sensores

Debido a sus características, las WSN pueden soportar un amplio rango de aplicaciones en diferentes ámbitos. Si se examina su flexibilidad, se puede decir que su potencial es infinito y que no hay un número determinado de aplicaciones, sino que, es una tecnología emergente y que la imaginación sería la única limitación. [12]

Para hablar de las aplicaciones que pueden tener las WSN, se hace necesario distinguir las mismas en varios grupos generales:

- *Automoción:* Este campo abarca desde la posibilidad de integrar una red completa de sensores dentro del vehículo para ofrecer información a los ocupantes, hasta una comunicación automática del coche con otras infraestructuras del ámbito vial como pueden ser señales, carreteras, semáforos, etc.
  Con respecto a las aplicaciones en automoción, cabe destacar el proyecto CAPSTONE de Ford desarrollado por la universidad estatal de Michigan [13] que pretende recoger los mayores datos posibles de su vehículo con el objetivo de mantener a los ocupantes informados en todo momento sobre el estado de funcionamiento de sus sistemas.
- *Control domótico:* orientadas principalmente al control energético de hogares o edificios2 permitiendo también adaptar el entorno a las necesidades y preferencias de las personas que se encuentran en su interior. [14]
- *Ambiental:* En este grupo se encuadran todas aquellas aplicaciones destinadas a la monitorización tanto de animales como de entornos naturales. Algunos ejemplos serían la monitorización de especies en extinción para proporcionarles un cuidado más intensivo o de grandes hectáreas de bosques para prever los incendios que se puedan ocasionar, glaciares y otros fenómenos afectados por el cambio climático, etc. [15] [16]
- *Control de almacenes:* Son aquellas que se utilizan para gestionar todos los materiales que se pueden encontrar en un almacén. [17]
- *Salud:* Se incluye todo lo relacionado con la monitorización y atención de los pacientes. Con esto, se puede evitar muchas veces que el paciente tenga que desplazarse a la consulta pudiendo avisar a un equipo médico en caso de que alguna de las señales caiga drásticamente.
  Este tipo de aplicaciones son muy útiles, sobre todo en personas mayores que requieren de un seguimiento exhaustivo. Con estas redes se les puede dejar una mayor libertad y privacidad en sus vidas al no obligarles a tener una persona cuidándoles continuamente. [18]
- *Control de procesos industriales:* Como se comentará en apartados posteriores, estas aplicaciones mejoran en gran medida el proceso industrial ya que, incluyendo una arquitectura SCADA se puede controlar todo el proceso desde un puesto centralizado. Además, puede haber sensores midiendo ciertos parámetros que de otra forma no podrían ser medidos o por lo menos no con la misma exactitud. [19]

[2] Tiene una relación estrecha con los hogares digitales

- *Militares:* Este fue, como la mayoría de estas tecnologías, uno de los motivos por los cuales se empezó a investigar en la materia, principalmente para tener conocimiento en todo momento de lo que está ocurriendo en el campo de batalla.
  Por ejemplo en 2006, Estados Unidos estuvo desarrollando un sistema que con una infraestructura WSN lograba enviar información procedente de sus fronteras al centro de operaciones donde era gestionada. [20]
- *Agricultura y ganadería:* Básicamente se refieren a las aplicaciones destinadas a mejorar la calidad y la producción agrícola, así como el control de posición de las cabezas de ganado. [21]
- *Seguridad y vigilancia:* Se puede considerar el más amplio de los grupos ya que incluiría todas las aplicaciones relacionadas, como su propio nombre indica, con la vigilancia de infraestructuras tanto públicas como privadas con el objetivo de proveerlas de mayor seguridad. [22]
- *Control de tráfico:* En este caso, los vehículos también podrían enviar información acerca de su dirección, situación y velocidad al centro de control de tráfico para que se permita realizar un control más exhaustivo que el que se realiza actualmente con cámaras. [23]
- *Estructuras:* Se enfoca principalmente en la monitorización continua de estructuras públicas o de gran tamaño que puedan ser destruidas en caso de desastres naturales como por ejemplo un terremoto. También en edificios antiguos para poder saber con anterioridad en caso de derrumbamiento.
  El ejemplo más famoso quizás es el del puente Golden Gate de San Francisco (California) [24] donde se instalaron 64 nodos que medían las vibraciones que pudieran ocurrir en el puente ya fueran producidas por vehículos o por condiciones atmosféricas.

1.4 Arquitectura SCADA en las Redes Inalámbricas de Sensores

1.4.1 Introducción

Con la Segunda Revolución Industrial (1880-1916) el proceso de producción cambió radicalmente al introducir máquinas que podrían sustituir las tareas que, hasta entonces, estaban desempeñando los trabajadores. Es obvio que, en esa época, no había ningún tipo de control de producción, por lo que la máquina funcionaba continuamente hasta que finalmente se estropeaba, lo que lógicamente implicaba la detención de la producción ante la necesidad de reparación de la misma.

Más adelante se empezaron a introducir mecanismos que ayudaban a controlar diferentes parámetros de la máquina. Éstos se basaban principalmente en sensores (de temperatura, presión, etc.) que determinaban el estado de funcionamiento de la misma. También hicieron su aparición paneles de botones con los que se podía manejar remotamente la máquina o alguna de sus funcionalidades y finalmente,

indicadores luminosos que hacían de señalización para el operario de la misma. [25]

En torno a 1960 se empezaron a introducir los Sistemas de Control Industrial (Industrial Control System, ICS) con el uso de PCs de software y de redes a través de las cuales se podrían realizar las mismas tareas que hacían los operarios in situ. [26]

Un tipo de ICS es SCADA (Supervisory control and data acquisition) cuya principal diferencia con los otros tipos radica en la posibilidad de utilizarlo en redes WAN (Wide Area Network).

En la Figura 5 se puede ver cómo sería un esquema básico de SCADA:

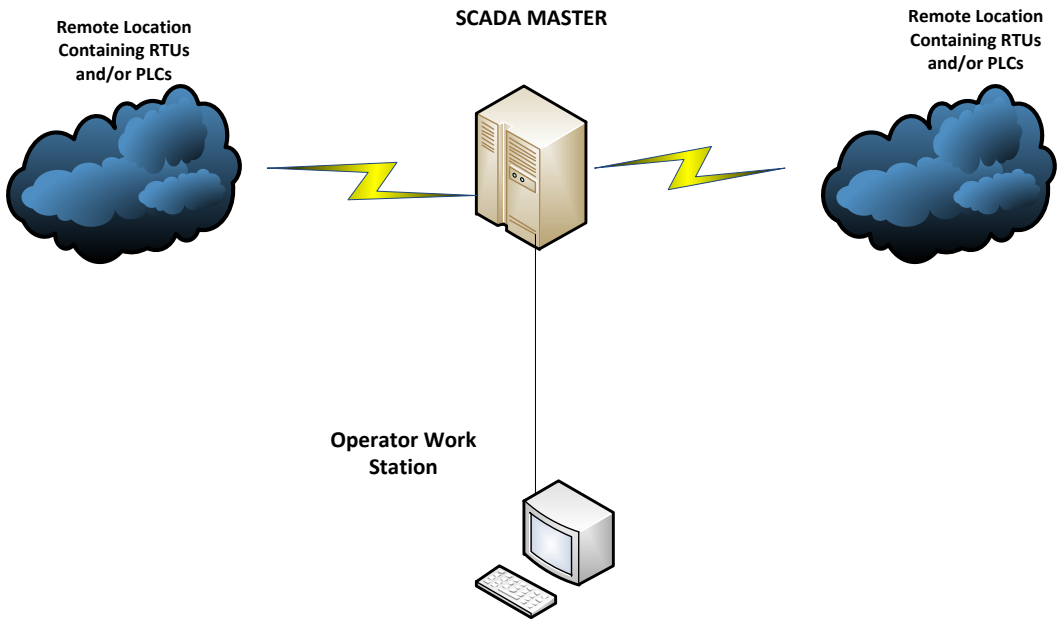


Figura 5. Esquema básico de SCADA [27]

1.4.2 Componentes

En un sistema SCADA intervienen diversos componentes que son los que, en conjunto, le dan sentido a un sistema de estas características.

Dispositivos de Control en planta

Pueden ser tanto sensores como actuadores y tienen poca capacidad de procesado. Están situados en los puntos estratégicos de la cadena de producción, es decir, en las máquinas, para poder así controlar diferentes parámetros. Como ejemplo se podría mencionar: medidores de temperatura, controladores de válvulas, controladores eléctricos, etc.

Ocurre a veces que estos aparatos no tienen el suficiente potencial de cómputo para poder, por ejemplo, traducir los datos recopilados al protocolo de comunicación que esté manejando el sistema SCADA en ese determinado caso y por tanto necesitan de otros elementos llamados los RTU (Remote Telemetry Units).

Por otra parte se encuentran los PLC (Programmable Logic Controller) que son los dispositivo que se encargan de la lógica, es decir, son los que en caso de necesidad, dependiendo de los parámetros obtenidos por los sensores, va a determinar qué tipo de acción es necesaria y en qué momento ejecutarla.

Al principio, era imposible guardar el algoritmo o programa que ejecutaban los PLC en los RTU y por eso eran dos dispositivos claramente diferenciados y cada uno con su tarea específica. Pero con el posterior desarrollo de estos sistemas, un mismo dispositivo puede ser capaz de realizar ambas labores, lo que en ocasiones ha generado cierta confusión ya que se ha tendido a que los dos nombres se usen para el mismo concepto. [26]

Red de Comunicaciones

Abarca todo lo relacionado con el propósito de comunicar los servidores centrales del sistema SCADA con los RTU. Históricamente se han usado diferentes soluciones como redes de cable o redes públicas de telefonía pero, con el desarrollo de las redes de larga distancia se ha podido realizar un mayor aprovechamiento de las posibilidades de este tipo de ICS consiguiendo un control remoto de los RTU a través de redes LAN (Local Area Network) y sobre todo WAN. [26]

Servidor Central

Pueden ser uno o varios servidores. También llamados MTU (Master Terminal Unit), [28] son los encargados de recibir los datos procedentes de los RTU que, según se puede apreciar en el ejemplo de la Figura 5, estarían conectados con los RTU a través de una red WAN. [26]

Por otro lado, los datos recopilados en el MTU han de ser interpretados por el operador del sistema y por ello el servidor/servidores centrales deben estar conectados con el ordenador/ordenadores de operarios, ofreciendo a su vez una interfaz “amigable” para la interpretación humana. [26]

Estaciones de Trabajo de los operadores

Se comunican directa y únicamente con el Servidor Central a través de una red que en el caso de la Figura 5 podría ser LAN y son los que actúan de clientes en la arquitectura Cliente-Servidor que se tiene en la aplicación distribuida de SCADA. [26]

Software

En SCADA existen diferentes posibilidades en cuanto al software. Es común que las empresas que se dedican a vender este tipo de soluciones, las acompañen con su propio "Set de Software" porque en este tipo de sistemas en los que actúan gran cantidad de dispositivos, se necesitan diferentes soluciones de software.

En un sistema SCADA los productos software necesarios son los siguientes [26]:

- Sistema Operativo del Servidor Central: el software puede ser Unix por ejemplo o cualquiera de los otros sistemas operativos para servidores.
- Sistema Operativo de la estación de trabajo del operador: generalmente coincidente con el del servidor central.
- Aplicación del Servidor Central: encargada de la comunicación con los RTU, generalmente también ofrece una interfaz de usuario.
- Aplicación de la estación de trabajo del operador: es la parte cliente de la aplicación que corre en el Servidor Central.
- Controladores del protocolo de comunicaciones: se encargan de la tarea de interpretar los datos que han sido enviados utilizando un determinado protocolo.
- Software para la administración de la red de comunicaciones: es específicamente para el control y la administración de la red.
- Software para el control de la RTU: como se ha hablado anteriormente se necesita un algoritmo de actuación para estos dispositivos y es proporcionado por dicho software.

1.4.3 Arquitectura

En los sistemas SCADA se pueden discernir diferentes fases o generaciones que determinarán el cómo ha ido evolucionando SCADA a medida que se desarrollaban ordenadores, RTUs y redes más potentes y con mayor capacidad de cálculo y velocidad.

Primera generación- Monolítica

Cuando se empezaron a instaurar los sistemas SCADA, no existían las redes de comunicación por lo que estos sistemas eran totalmente independientes con el RTU que se quería manejar.

Posteriormente, con la introducción de las redes WAN, se pasó al control remoto de las RTU pero la arquitectura seguía siendo tal que el equipo donde estuviera instalado SCADA se comunicaba con cada RTU con una

red WAN a través de un protocolo de comunicaciones que del que era propietario la empresa que vendía el SCADA. [29]

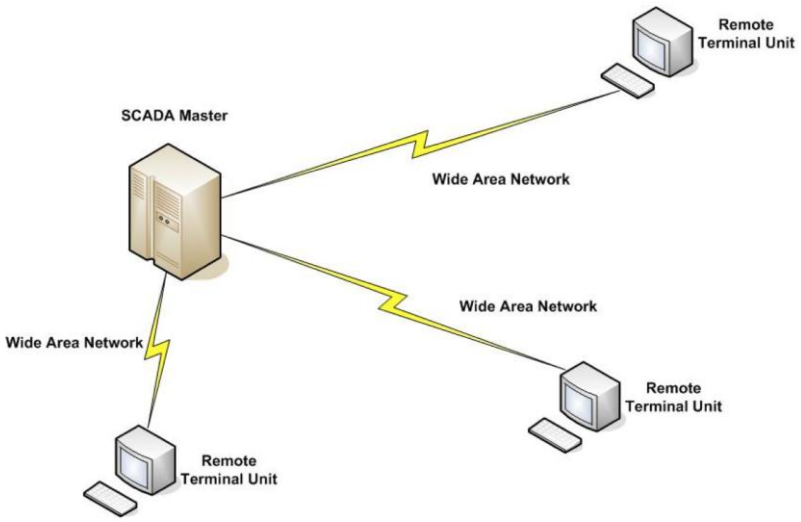


Figura 6. Primera generación de SCADA [26]

Segunda generación- Distribuida

Esta segunda generación está caracterizada por la inclusión de la red LAN en la arquitectura de SCADA y con ello la posibilidad de procesamiento en tiempo real y la compartición de datos entre los diferentes componentes del sistema. Con el avance tecnológico también se pudieron obtener estaciones de trabajo más pequeñas y simplificadas por lo que se separó la parte lógica del interfaz que se le ofrecía al operador.

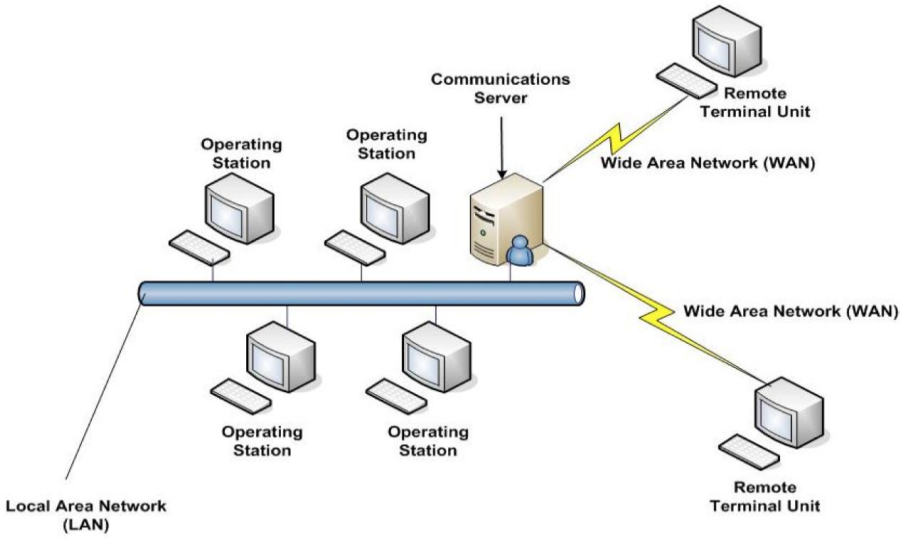


Figura 7. Segunda generación de SCADA [26]

Los protocolos que se utilizaban en la red LAN eran propietarios de la empresa vendedora, se ofrecía tiempo real pero carecía de cierta flexibilidad. Sin embargo para la comunicación entre los servidores y los RTUs seguían la topología de la anterior generación a través de redes WAN. [29]

Tercera generación- De red

La característica principal de esta generación es la inclusión en los sistemas SCADA de protocolos abiertos, no solo propietarios de la empresa vendedora, por lo que a partir de aquí se eliminaron todas las limitaciones y se proporciona una gran flexibilidad en cuanto a la arquitectura.

Esto también incluye la posibilidad de utilizar una red IP como medio de comunicación y por tanto Internet, con lo que se podrían controlar las RTU incluso desde cualquier parte del globo terráqueo. [29]

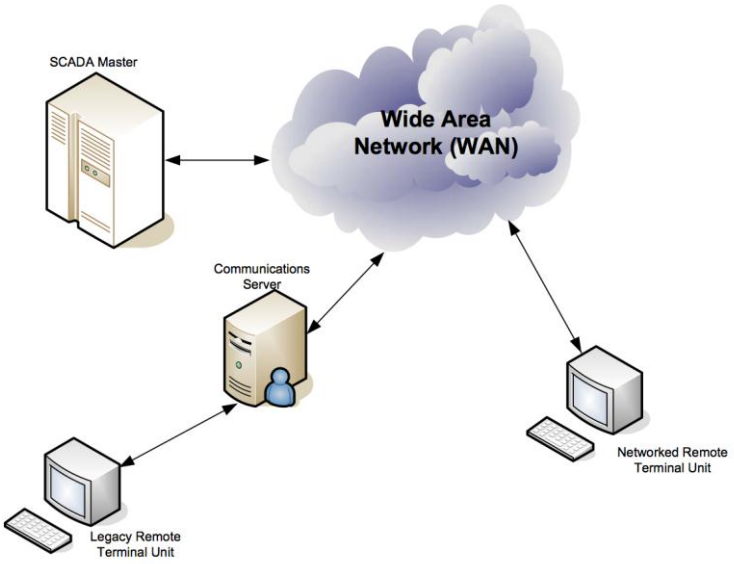


Figura 8. Tercera generación de SCADA [26]

Cuarta generación- IoT

Según Yvette E. Gelogo et al, la tendencia apunta a que cada vez se utiliza más el protocolo IP en las comunicaciones SCADA. También es cierto que las WSN se están volviendo cada vez más importantes ya que se usan en el "Internet of Things".

En esta generación la principal característica que se quiere implementar en el sistema SCADA es la posibilidad de que, con la utilización del protocolo IPv6, todos los RTU puedan obtener información de forma remota de cualquier sensor, con la tecnología Wireless,.

El papel del Servidor Central en este escenario sería el mismo salvo que además se tendrían los datos en tiempo real acerca de qué sensor está conectado a qué RTU conociéndose así en todo momento la información de la topología de la red, lo que permite al servidor saber cómo tiene que encaminar los datos que quiera enviar a uno de los sensores. [28]

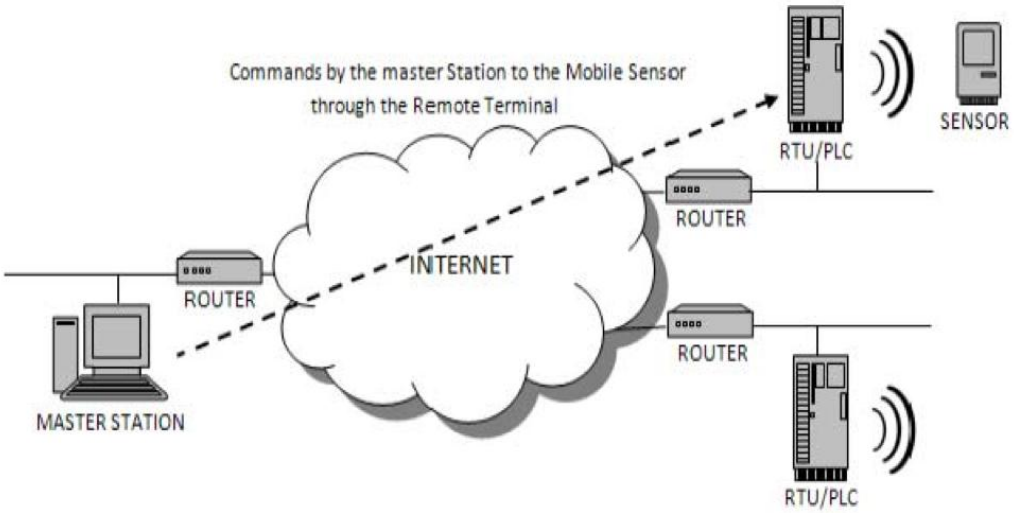


Figura 9. Cuarta generación de SCADA [28]

2 Seguridad en redes WSN

2.1 Introducción

Una WSN (Wireless Sensor Networks) , como cualquier otra red, pueden tener vulnerabilidades y sufrir amenazas o ataques, por tanto es importante proveer a dichas redes de cierto nivel de seguridad, es decir, implantar servicios de seguridad y por consiguiente mecanismos de seguridad efectivos capaces de mitigar cualquier ataque o amenaza que pueda inferir en el servicio que está proporcionando dicha red. [27]

Como se ha visto anteriormente las WSNs tienen multitud de aplicaciones y su flexibilidad permite la utilización de las mismas para proporcionar una enorme cantidad de servicios. Por lo tanto la seguridad que se debe proveer a cada servicio, se debe analizar particularmente ya que, por ejemplo, no es lo mismo si los datos que manejan los nodos son datos privados de personas como puede ser su presión sanguínea, su colesterol, etc; que si tratamos esos mismos datos pero pertenecientes a un animal. También hay que tener muy en cuenta el ámbito en el que se está aplicando el servicio de seguridad porque, por ejemplo, se tendrán que tener más mecanismos de seguridad en aplicaciones militares, que en el resto. [30]

Por otro lado como dicen Luis E. Palafox et al: “evidence the easiness of perpetrating several types of attacks due to the extreme resource limitations that wireless sensor networks are subjected to” [31], lo que quiere decir que al ser unas redes donde existen muchas limitaciones, deben de tener un trato a medida en cuanto a mecanismos y servicios de seguridad.

Por lo anterior, se han de definir las posibles amenazas y vulnerabilidades de cada sistema en particular, para así poder aplicar servicios y mecanismos de seguridad siempre y cuando no se perjudique al servicio que se quiere ofrecer con la WSN.

2.2 Requerimientos de seguridad

Dado que las WSN son un tipo de redes especiales por la información que pueden transportar, el nivel de procesamiento de sus nodos, la duración de la batería, la limitada capacidad de memoria, etc., no se puede aplicar directamente un entorno de seguridad pensado para redes de ordenadores pero sí necesitan ciertos requerimientos de seguridad y de continuidad para poder garantizar los servicios que se pretenden proporcionar. [32] Estos requerimientos dependen de qué tipo de servicio provee la red pero en general se resumen en la Figura 10. [30]

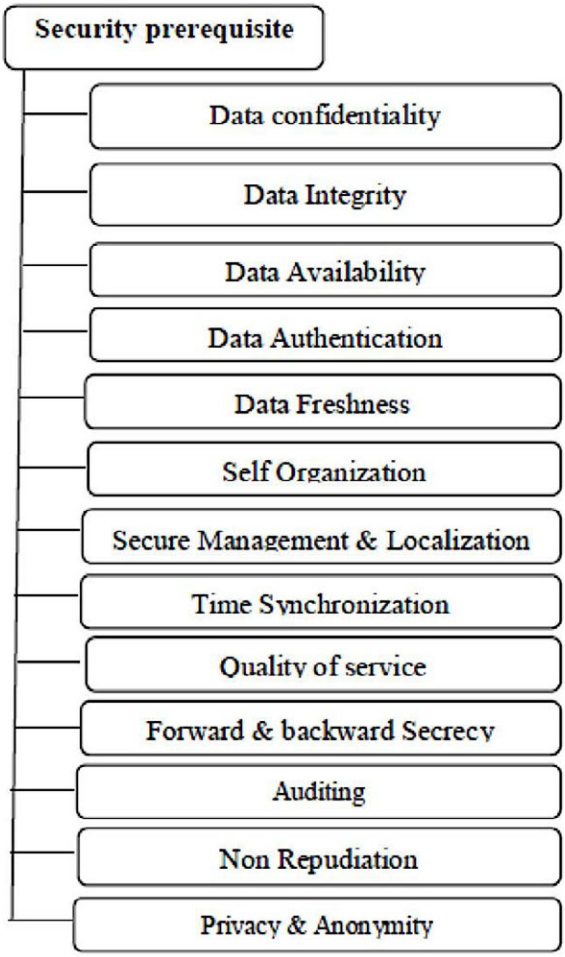


Figura 10. Prerrequisitos de seguridad para las WSN [27]

Data confidentiality (Confidencialidad en los datos): Garantiza que los datos que se intercambian los nodos no puedan ser leídos o interpretados por terceros con lo que se asegura confidencialidad en la información. [27]

Data integrity (Integridad en los datos): Permite asegurar que los datos no han sido manipulados por un tercero, con lo que el receptor sabe que la información que le llega no se ha podido modificar durante el tránsito por la red. [27]

Data availability (Disponibilidad de los datos): Con este prerrequisito se puede asegurar la continuidad de la red y que en todo momento el servidor central estará recibiendo datos de cada uno de los nodos de la WSN. [27]

Data freshness (Datos recientes): Confirma que los datos que se envían no son repeticiones de mensajes antiguos, lo que podría causar confusión en la red. [27]

Self Organization (Auto-Organización): Para asegurar la continuidad, los nodos deben poder recuperarse ante adversidades como por ejemplo que a uno de ellos se le acabe la batería. Ante esto se precisa dinamismo en los protocolos de señalización entre ellos para poder reorganizarse y no perder el nivel de seguridad que se tenía en el anterior estado [33]

Secure management & Localization (Localización y mantenimiento seguro): Es necesario en una red de este tipo saber en todo momento cuantos sensores tiene cada nodo, donde están, y si están funcionando o no. Aparte es imprescindible el reparto de claves para una correcta encriptación que proporcionarán otros servicios, pero este reparto se debe hacer también en un entorno de seguridad. Así mismo, debe quedar protegida la información de routing, pilar muy importante para el buen funcionamiento de la red y que queda fuera del payload (carga útil). [27]

Time Synchronization (Sincronismo): Como en cualquier sistema distribuido, este es un requerimiento importante ya que en muchas ocasiones este tipo de redes necesitan que la información sea instantánea como por ejemplo en el escenario en el que estén implicados bomberos, policía, etc. [27]

QoS (Calidad de Servicio): Afecta tanto a aspectos relacionados con la aplicación en sí misma, que incluiría la precisión de medida de los sensores, el número mínimo de sensores activos…, y en general todo lo relacionado con una Calidad de Servicio desde el punto de vista Usuario-Aplicación como a la red en cuestiones referentes a la forma en que se entregan los datos desde los nodos al servidor central y que dependerían de ciertos factores como: [27]

- Si la aplicación es interactiva, es decir, si desde el servidor se puede mandar información a los nodos para modificar su comportamiento o pasiva, si sólo recibe información que ha de ser interpretada.
- Si se permite cierto retraso en la entrega de los paquetes.
- Si se trata de una aplicación punto a punto, etc.

Forward and Backward Secrecy: Este concepto se refiere a que, en el caso de "forward secrecy", cualquier sensor de la red no debe ser capaz de leer mensajes una vez haya abandonado la misma. Por el contrario el de "backward secrecy" significa que tras la incorporación de un sensor en la red, éste no debe ser capaz de saber la clave de sesión que tenía el anterior. [27]

Auditing (Auditoría): Es la comprobación de que la red puede ser autosuficiente en el caso de que se necesite reemplazar algún componente de la misma con mecanismos de virtualización. [27]

Non Repudiation (No repudio): En el caso de las WSN implicaría que los nodos no puedan negar el envío de un determinado dato al servidor y viceversa. [27]

Privacy and anonymity (Anonimato y Privacidad): Dado que en este tipo de redes pueden estar implicados seres humanos y de los cuales se obtienen datos que pueden ser o no privados se debe mantener tanto el anonimato de la persona con la que se está tratando o de la que se está obteniendo información como la privacidad de los datos recopilados. [27]

2.3 Restricciones y vulnerabilidades

Las Redes Inalámbricas de Sensores son un tipo especial de red, y como tal tienen una serie de limitaciones que las hacen ser más vulnerables en algunos aspectos que las redes convencionales de ordenadores. Las principales son las siguientes:

- *Recursos limitados*
    - *Limitación en memoria y en almacenamiento:* En los sensores, el tamaño puede variar desde el tamaño de una caja hasta el de una moneda y por tanto la memoria y el almacenamiento pueden llegar a ser escasos kilobytes, por lo que las líneas de código de la parte de seguridad tienen que ajustarse a el espacio de memoria que esté designado para ello. Un ejemplo sería el sensor TelosB que es uno de los más típicos y que solo consta de 10K de RAM y de 48K de memoria de programa.
    - *Limitación en energía:* Esta es una de las cuestiones más importantes porque en la mayoría de las ocasiones los sensores se despliegan con el objetivo de no tener que reemplazar las baterías o volver a cargarlas cada cierto tiempo porque muchas veces saldría muy caro. Por tanto se intenta que la batería dure el máximo tiempo posible y para ello los sistemas de seguridad tienen que estar pensados para que consuman la menor energía posible de la batería y así poder alargar más la vida del sensor. Los principales consumos de energía que se producen en los sensores son en funciones de seguridad (encriptado, desencriptado, firmado de datos, verificación de firma), la transmisión de datos de seguridad (inicialización de vectores para encriptar y desencriptar), y el almacenamiento de los parámetros de seguridad y claves de manera segura (almacenamiento de claves criptográficas).
- Comunicación no fiable
    - *Transferencia no fiable:* Los paquetes generalmente viajan por redes inalámbricas, lo que puede ocasionar que se produzcan errores en los mismos o que algunos sean descartados en los nodos por la congestión, por lo cual es necesario la inclusión de un mecanismo para manejar estos errores por parte del protocolo, para poder prevenir que se pierdan paquetes importantes como son los que contienen información de seguridad (Por ejemplo claves criptográficas).
    - *Conflictos:* Aunque el canal de comunicación sea totalmente fiable, también pueden ocurrir problemas de conflictos, sobre todo en las redes muy congestionadas.

- *Latencia:* Si se trata de una red con enrutamiento basado en "multi-hop" se puede producir una gran latencia en la entrega de paquetes y por consiguiente fallos en la sincronización que es un pilar para poder mantener un sistema de seguridad a tiempo real.

- Desatención continuada
  - *Vulnerabilidades físicas:* Las Redes Inalámbricas de Sensores están pensadas para ser desplegadas en lugares donde los nodos no tienen por qué estar protegidos en sentido físico por lo que fenómenos naturales como la lluvia, podría dañarlos o incluso un atacante podría capturar un nodo e intentar realizar un ataque que afecte a toda la red, lo que resulta especialmente dañino en el caso, por ejemplo, de las aplicaciones militares.
  - *Administración remota:* Los nodos se controlan generalmente de forma remota y muchas veces se antoja prácticamente imposible el acceder a ellos de forma física.
  - *Descentralización de la administración:* Una red de sensores debería de ser descentralizada porque así se le puede dar una mejor viabilidad. [34]

Como se puede observar en la Figura 11, existen también otras clasificaciones de las vulnerabilidades que pueden tener las Redes Inalámbricas de Sensores. [27]

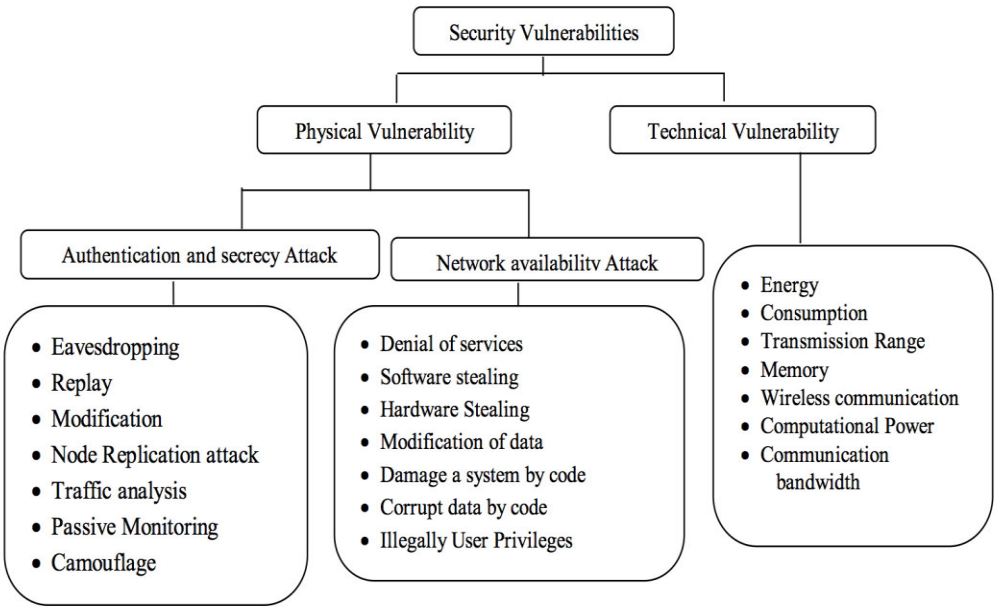


Figura 11. Vulnerabilidades en las Redes Inalámbricas de Sensores. [27]

2.4 Evaluación de esquema de seguridad

Y. Wang et al sugieren una serie de pautas a seguir para poder evaluar si un entorno de seguridad es apropiado o no para una WSN. Para ello dicho sistema de seguridad ha de tener las siguientes características: [33]

- *Seguridad:* Deberá responder a un esquema que tenga en cuenta las necesidades particulares de cada red y que le proporcionen protección ante sus vulnerabilidades.
- *Resiliencia:* Los nodos deben mantener la seguridad en la red aun en el caso de que alguno esté comprometido.
- *Energía:* se debe aprovechar al máximo el tiempo de vida de las baterías de los nodos por lo que el consumo ha de ser eficiente.
- *Flexibilidad:* Para que en el caso de cambios en la tipología o de utilización de diferentes esquemas se pueda utilizar el mismo sistema de claves.
- *Escalabilidad:* Siempre sin comprometer la seguridad de la red.
- *Tolerancia al fallo:* La red debe poder responder con el mismo nivel de seguridad aunque uno de los nodos o varios fallen.
- *Auto reparación:* Es vital para la red que los mecanismos de seguridad se puedan reestructurar automáticamente en el momento que un nodo falle.
- *Garantía:* consiste en que la red sea capaz de divulgar información a los usuarios y que éstos puedan elegir diferentes parámetros concernientes a la QoS que quieren implementar acorde con el esquema de seguridad.

2.5 Tipos de ataques

Las Redes Inalámbricas de Sensores pueden sufrir diversos tipos de ataques. En muchos casos son ataques que podrían ocurrir en otro tipo de redes pero en otros se deben a las limitaciones propias de las WSN según se ha comentado en el apartado 2.3. Generalmente los más comunes son los ataques de Denegación de Servicio (DoS) pero también se dan casos de violación de la privacidad, ataques físicos, de análisis de tráfico, etc. [33]

Cada ataque que existe en la actualidad y los posibles ataques que aparezcan para estas redes, se podrían clasificar en grandes grupos dependiendo de diferentes parámetros como son las capas del modelo OSI (de las cuales están: Física, Enlace, Red, Transporte y Aplicación) o dependiendo de si es un ataque Activo o Pasivo de acuerdo con Sunil Gupta *et al.* [27] como se muestra en la Figura 12.

Sin embargo también son posibles otros tipos de clasificaciones como por ejemplo:

- Ataques internos o externos: desde fuera si los causa un nodo que no pertenecía a la red o desde dentro si es el caso de un nodo que está dentro de la red pero que se comporta de forma no autorizada. [33]

- Ataques según su procedencia: de nodos si es causado por uno o varios nodos a la vez o ataques desde ordenadores o máquinas más potentes que los propios nodos. [33]

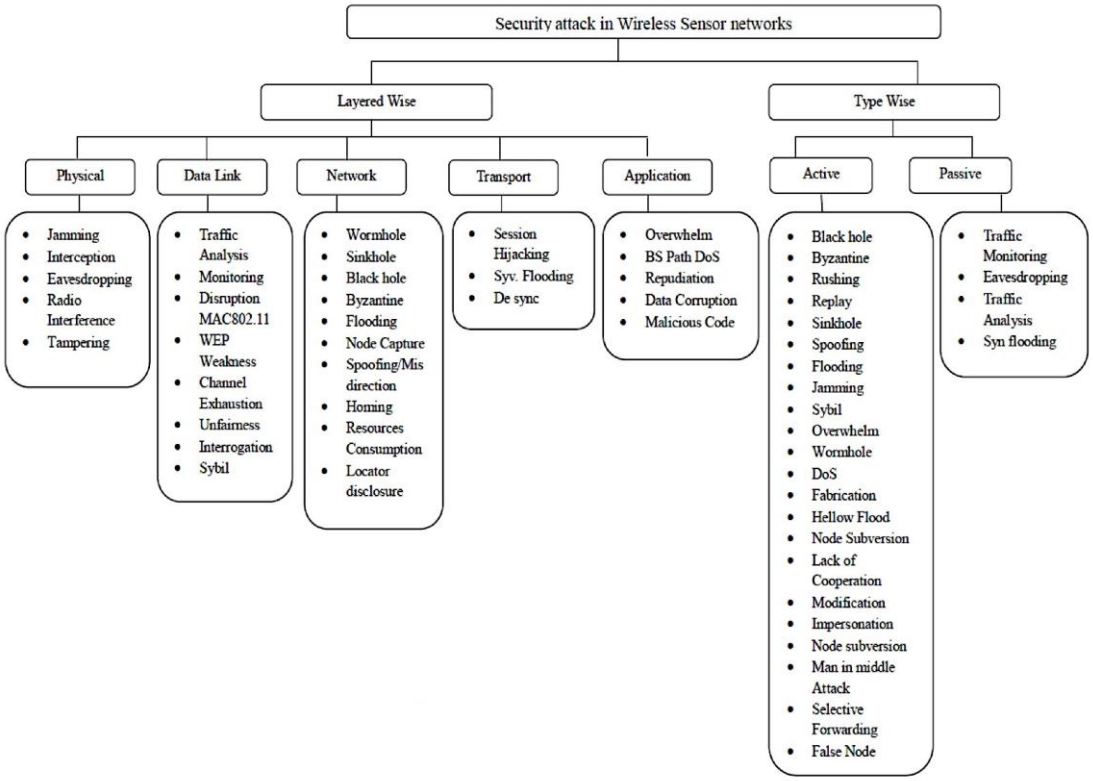


Figura 12. Ataques en una red inalámbrica de sensores. [27]

Pero sin duda el método más común de clasificación es atendiendo al objetivo que tengan, es decir, el servicio que pretenden irrumpir (Por ejemplo, la DoS, Deny of Service, la violación de la privacidad, producir fallos de sincronización, cambiar información de los paquetes, etc). Se pueden destacar por tanto los siguientes grupos:

- *Ataques de denegación de servicio:* se podría definir como "any event that diminishes or eliminates a network's capacity to perform its ex- pected function" de acuerdo con A. D. Wood y J. A. Stankovic [35]. Teniendo en cuenta los tipos de aplicaciones de las WSN, se puede considerar este tipo de ataque como uno de los más peligrosos, sobre todo en situaciones en las que se necesita de una constante monitorización como puede ser el caso de una alarma de incendios en un edificio.

  Dentro de este grupo se pueden distinguir ataques característicos de DoS como son el Jamming (que se basa en producir interferencias a nivel físico de la red con radiofrecuencias que cubren el rango de frecuencias que está utilizando el protocolo de transmisión en la red) o el Flooding (que es un ataque a nivel de transporte caracterizado por enviar a un nodo muchos

paquetes de intento de conexión produciendo un desbordamiento del mismo). [35] [33]

- *Ataque Sybil:* Este es un tipo de ataque que se basa en que un nodo malicioso pueda utilizar diferentes identidades en la red. Está pensado para redes que almacenan datos de forma distribuida como son, por ejemplo, las clásicas P2P (Peer to Peer). En el caso de las WSN distribuidas, el nodo malicioso, al tener múltiples identidades, haría que todos los datos fueran enrutados hacia él y por tanto toda o la mayoría de la información que estaría en tránsito en la red.

- *Ataques de análisis de tráfico:* Consiste en atacar a la estación base que recoge la información de los sensores típicos de redes WSN o simplemente deshabilitarla, con lo que la red estaría prácticamente desmontada. Por ejemplo, Deng et al demostraron, con dos ataques diferentes, que se puede detectar cuál es la estación base e incluso interceptar los paquetes que le llegaban. [36]

- *Ataques de réplica de nodos:* El atacante puede introducir en la red un nodo con los mismos identificadores y claves que otro ya existente lo que produciría errores en el enrutamiento, pérdidas de paquetes, etc. También otra posibilidad sería sustituir el nodo original por la réplica a la que el atacante tendría acceso.

- *Ataques contra la privacidad:* Entre los muchos escenarios de aplicación de este tipo de redes, se encuentran aquellos en los que datos de personas fluyen en la red. El rápido desarrollo del Internet de las Cosas, aunque aporte muchas ventajas, hará que multitud de datos personales estén expuestos, lo que podría suponer un mayor riesgo de ataques que afecten a la privacidad. Por ejemplo, C. Ozturk et al exponen el caso de que, un cazador que ataque una red que esté monitorizando la posición de algún tipo de animal con fines estadísticos únicamente, podría saber exactamente donde se encuentra en ese momento el animal y por tanto practicar caza furtiva. [37]

- *Ataques físicos:* Este grupo es el más claro de todos. Muchas veces las redes están desplegadas en entornos hostiles y prácticamente desatendidas por lo que son susceptibles a ataques físicos que lleven a su destrucción. [38]

2.6 Contramedidas

Anteriormente se han expuesto los tipos de ataques que pueden afectar al funcionamiento normal de una Red Inalámbrica de Sensores. Para paliarlos es

necesario disponer de contramedidas eficientes, entendidas como mecanismos necesarios para mitigar la amenaza que conllevan esos ataques.

Al igual que en el caso de los ataques que afectan a las WSN, se pueden hacer varias clasificaciones de los mecanismos. En la recomendación X.800 de la ITU (International Telecommunicacion unit) [39] se expone una clasificación por capas basándose en el modelo OSI (Open System Interconnection). No obstante, puede haber mecanismos que se implementen en una determinada capa y que también puedan contrarrestar ataques de otras capas o que un solo mecanismo pueda cubrir amenazas procedentes de varios ataques.

En los siguientes apartados se describen los mecanismos por capa de implementación de acuerdo con la recomendación X.800.

2.6.1 Capa física

En la capa física la principal contramedida que se puede implementar es el cifrado de la información transmitida. Se realiza con un dispositivo que opera de forma transparente con el objetivo de proporcionar el servicio de confidencialidad en los datos.

2.6.2 Capa de enlace

En este caso el cifrado se produciría un paso antes de las funciones de transmisión involucradas en esta capa y en el lado receptor, después de las funciones de recepción típicas.

El cifrado está estrechamente relacionado con el protocolo usado en la capa ya que los mecanismos de seguridad que se puedan implementar utilizan las funciones de la capa.

2.6.3 Capa de red

En esta capa es donde se produce el encaminamiento. Por tanto todo mecanismo implementado está asociado con el protocolo encargado del acceso a la subred y las operaciones de relevo y encaminamiento que se esté utilizando. Por lo general se pueden aplicar los siguientes mecanismos:

- Mecanismos de intercambio de contraseñas protegidos y mecanismos de firma que proporcionan el servicio de autenticación de entidad par.
- Mecanismos de cifrado de firma que aseguran autenticación del origen.
- Mecanismos de control de acceso que certifican el servicio de control de acceso.
- Mecanismo de cifrado y control de encaminamiento que proveen el servicio de confidencialidad en modo con y sin conexión.
- Mecanismos de relleno de tráfico que combinados con servicios de confidencialidad y control de encaminamiento proporciona el servicio de confidencialidad de flujo de tráfico.

- Mecanismos de integridad de los datos que prestan el servicio de integridad en modo con y sin conexión.

2.6.4 Capa de transporte

Los mecanismos actúan de forma que cada conexión de la capa de transporte pueda aislarse de todas las demás conexiones de transporte. Los mecanismos que actúan en dicha capa son del mismo tipo que los de la capa de red.

2.6.5 Capa de sesión

En la capa de sesión no se proporcionan servicios de seguridad, por tanto, no se puede especificar ningún mecanismo para esta capa.

2.6.6 Capa de presentación

Los mecanismos de la capa de presentación suelen actuar en conjunto con los mecanismos de seguridad de la capa de aplicación.

- Mecanismos de cifrado que proporcionan los servicios de autenticación de entidad par, confidencialidad en modo con y sin conexión, confidencialidad de campos seleccionados y confidencialidad de flujo de tráfico.
- Mecanismos de firma que proveen el servicio de autenticación de origen.
- Mecanismos de integridad de datos que aseguran los servicios de integridad en modo con y sin conexión.
- Mecanismos de comprobación que, en combinación con los de firma y de integridad de datos pueden proporcionar el servicio de no repudio tanto del origen como de la entrega.

2.6.7 Capa de aplicación

Muchos de los servicios que se pueden aportar en esta capa utilizan los mecanismos de capas inferiores o combinaciones de ellos con los propios mecanismos de la capa de aplicación. Los mecanismos que se pueden implementar en la capa de aplicación son:

- Mecanismos de relleno de tráfico que combinado con un servicio de confidencialidad de una capa inferior proporciona un servicio limitado de confidencialidades del flujo de tráfico.
- Mecanismos de control de acceso que en combinación con mecanismos del mismo tipo de capas inferiores aseguran el servicio de control de acceso.

DESARROLLO FUNCIONADO

3 TSES (Technological Solutions Expert System)

3.1 Marco en el que se ubica

El Sistema Experto TSES se encuadra dentro de un sistema global llamado PDPS-IoT (Provisión de Políticas de Seguridad en el Internet de las Cosas). En su conjunto, el sistema global tiene como objetivo determinar los mecanismos y servicios de seguridad que puedan ser más efectivos para diversas arquitecturas de red pero que a la vez se puedan implementar teniendo en cuenta las características específicas.

El sistema general, está pensado para trabajar no solo con redes convencionales de ordenadores sino también para Redes Inalámbricas de Sensores del "Internet de las cosas", así como futuros tipos de redes que canalicen servicios de Internet de las Cosas.

Se compone de cuatro subsistemas expertos enlazados en serie donde la salida de cada uno correspondería con la entrada del siguiente como se puede ver en la Figura 13.

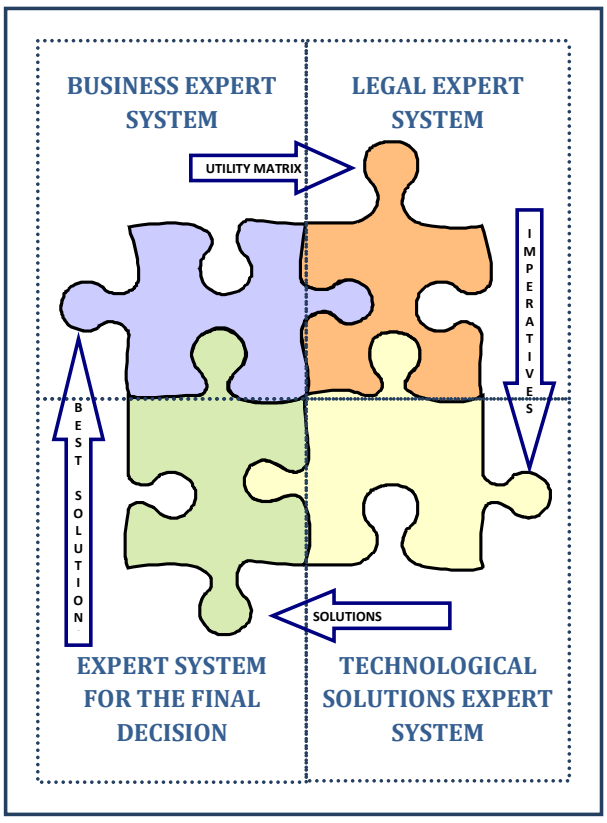


Figura 13. Visión general del Sistema PDPS-IoT

A continuación se describe la funcionalidad particular de cada sistema:

- *Business Expert System (BES):* Proporciona una matriz de utilidad del servicio final.
- *Legal Expert System (LES):* Aporta los imperativos legales y normativos.
- *Technological Solutions Expert System (TSES):* Proporciona soluciones de políticas de seguridad.
- *Final Decision Expert System (FDES):* Selecciona los servicios de seguridad que deben aplicarse de entre las posibilidades ofrecidas por el sistema anterior.

En la Figura 14 se puede ver dónde está encuadrado el sistema TSES que es el objeto de estudio en este proyecto, así como los flujos de datos entre los diferentes sistemas expertos.

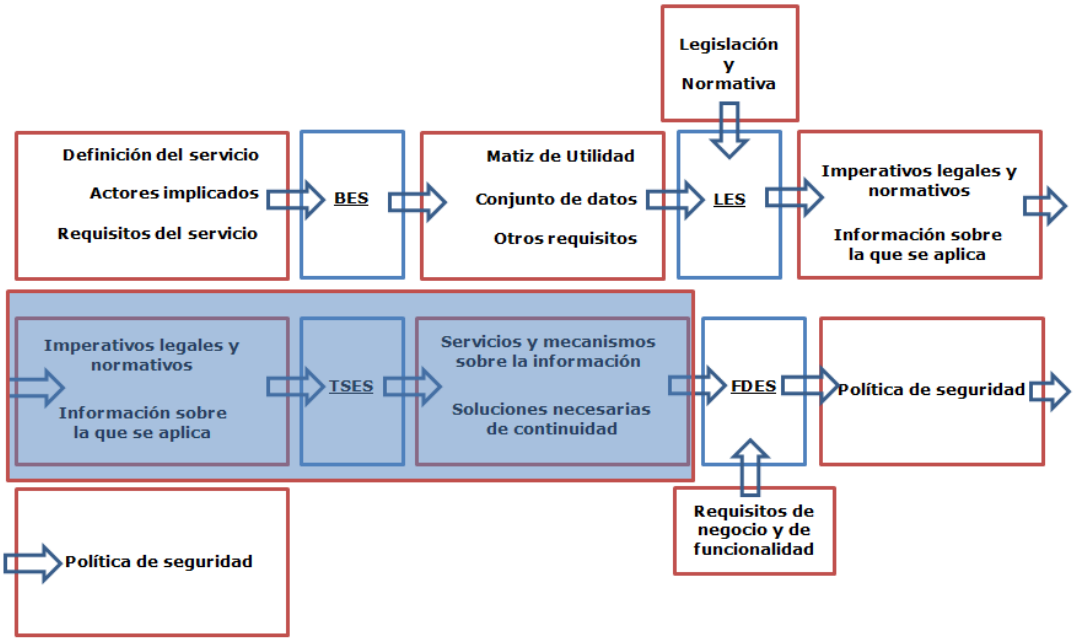


Figura 14. Visión específica de PDPS-IoT

En una visión más general, el sistema PDPS-IoT servirá de apoyo a la plataforma que se está desarrollando en el proyecto AWARE llevado a cabo por el CITSEM [40].

AWARE es la plataforma encargada de, a través de mensajes de configuración, implantar en un sistema SCADA la política de seguridad necesaria. Para ello AWARE necesita consultar al sistema PDPS-IoT cuál es la política de seguridad más apropiada en cada caso. Este proceso de comunicaciones se puede observar en la Figura 15.

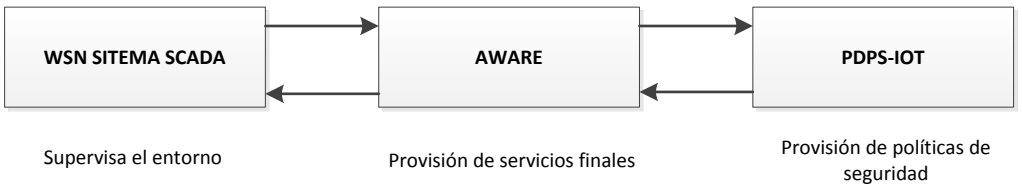


Figura 15. Servicio genérico de "Internet de las Cosas"

3.2 Objetivo concreto de TSES

TSES es un sistema experto que tratará las entradas que consisten en información acerca del tipo de red que se quiere securizar, y en un conjunto de imperativos legales y normativas de seguridad, así como en los datos que se han de proteger.
Mediante el procesamiento de la información anterior, se obtendrán los servicios y mecanismos de seguridad que se necesiten implementar en la red para el caso de uso del servicio concreto del "Internet de las Cosas" que se quiere ofrecer.
Para conseguir su objetivo, TSES tendrá en cuenta también los datos consultados en una base de conocimientos que debe ser completada por expertos de seguridad y actualizada con los últimos avances en lo que a seguridad en redes se refiere.
En la Figura 16 se puede observar el funcionamiento general del sistema experto junto con las informaciones de entrada y salida para el mismo.

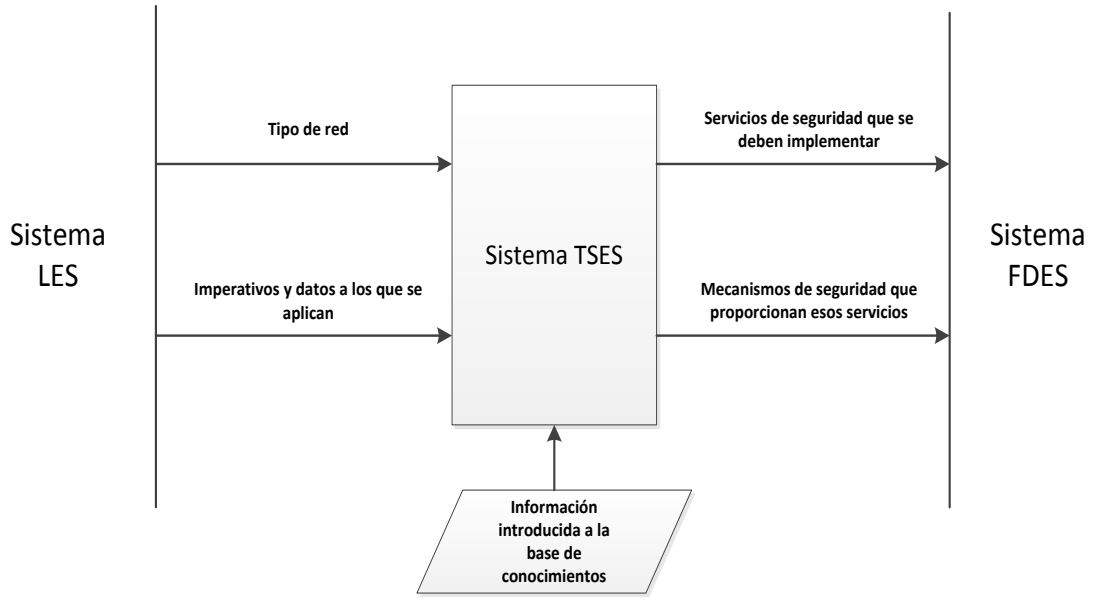


Figura 16. Visión general del sistema

3.3 Visión general de la aplicación

Desde un principio, se decidió como requisito indispensable de funcionamiento del subsistema TSES que deberían cumplirse dos características principales: que se pudiera acceder de forma remota para que así, diferentes expertos de seguridad

autorizados puedan rellenar la base de conocimientos y también que permitiera el acceso de forma concurrente para facilitar a los usuarios el uso del sistema en cualquier momento.

Por lo tanto se ha realizado una aplicación basada en web que tendría una arquitectura como la que se puede observar en la Figura 17.

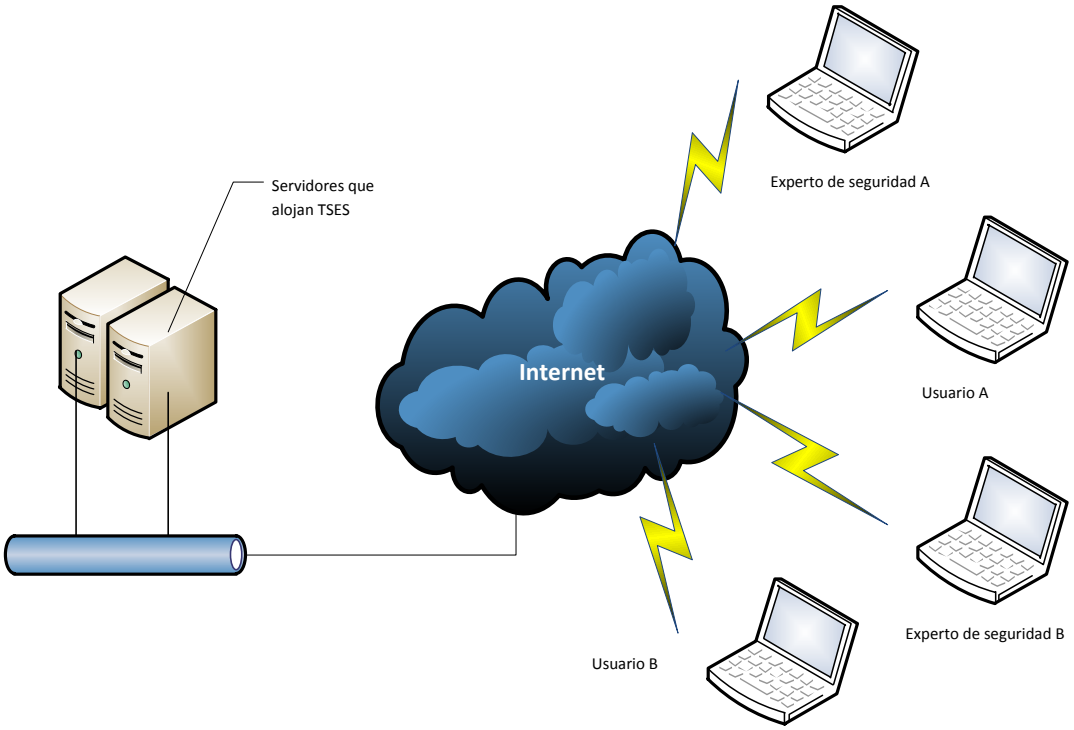


Figura 17. Arquitectura de red TSES

Se ha intentado realizar una arquitectura orientada, sobre todo, a un manejo sencillo e intuitivo por parte de un usuario no experimentado. Para ello se ha utilizado un interfaz muy simple que a través de menús permite la navegación por toda la aplicación web que, además, es totalmente independiente del navegador que se esté utilizando.

4 Nivel de desarrollo

4.1 Herramientas de desarrollo

Una vez decidido que se trataría de una aplicación web, se concluyó que se necesitarían, al menos un servidor para alojar la aplicación en sí y otro que hace las veces de servidor de bases de datos.

Acerca del software, se ha escogido para la parte de "back-end" la tecnología Java Server Faces (JSF) para la parte de aplicación mientras que para las transacciones con la base de datos se utiliza Java Persistence API (JPA), ambas ubicadas dentro del grupo Java Enterprise Edition (JEE). También cabe mencionar la necesidad de un interfaz funcional orientado al usuario y para este apartado de "front-end"[3] se han utilizado las tecnologías XHTML (eXtensible HyperText Markup Language), JavaScript y CSS3.

Por tanto la estructura de capas de los servidores utilizando la tecnología JEE proporcionada por Oracle quedaría como en la Figura 18.

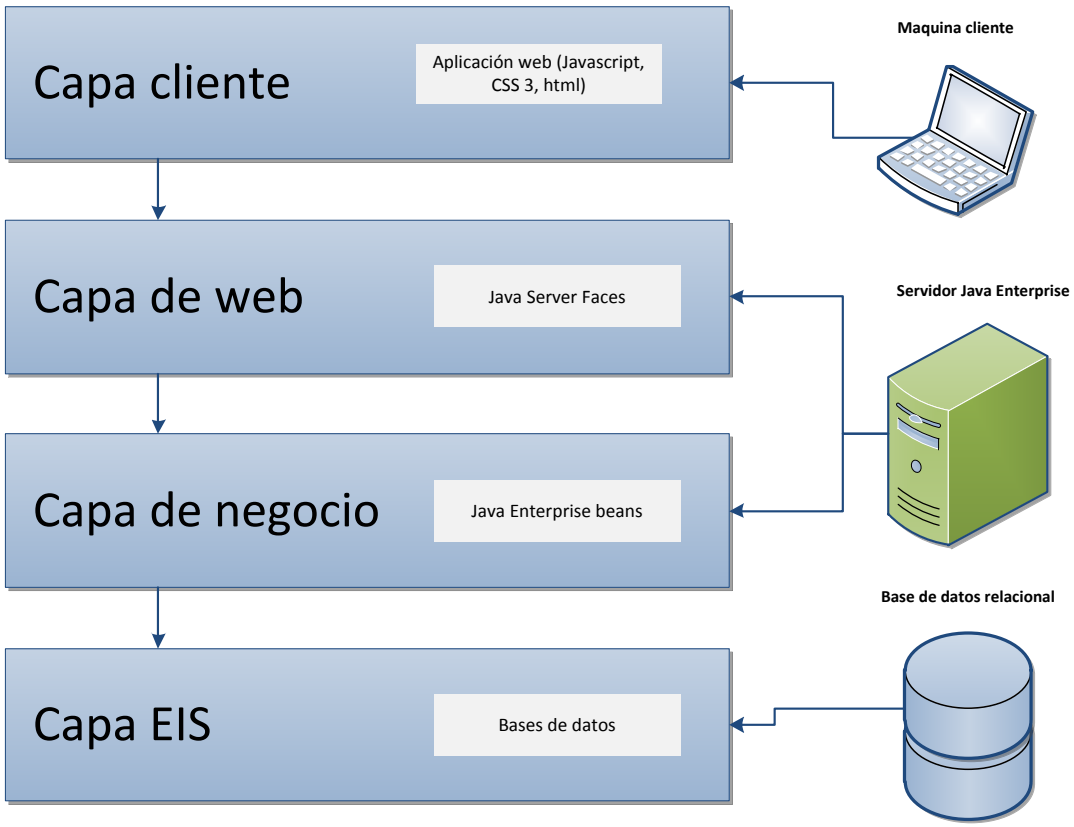


Figura 18. Arquitectura por capas de la aplicación

[3] El "front-end" es la parte del software que interactúa con el o los usuarios y el "back-end" es la parte que procesa la entrada desde el "front-end" y generalmente se sitúa en el lado del servidor.

4.1.1 Java Enterprise Edition

Se puede decir que Java Enterprise es un conjunto de tecnologías proporcionado por Oracle que se usa para la lógica de negocio de aplicaciones Enterprise. Está orientada a la interactuación entre diferentes tipos de software Enterprise y a ofrecer una mayor velocidad de desarrollo y construcción de aplicaciones web. Para ello, existe una API que reduce la complejidad que puede haber en un entorno de desarrollo de aplicaciones Enterprise a través de descriptores XML, anotaciones, código Java, etc. [41]

Para la implementación del sistema TSES, se ha recurrido a la versión más moderna de dicha tecnología que es Java EE 7 y que como se verá en apartados posteriores influye en aspectos como la versión mínima de servidores que se deben de utilizar.

Como se ha mencionado anteriormente, en la aplicación aquí expuesta, se han utilizado dos tecnologías pertenecientes al grupo de Java EE:

- Java Server Faces: Se trata de un framework perteneciente a la especificación JSR-314 [42] que permite al desarrollador construir aplicaciones basadas en web.
  Se compone de dos elementos principales: una API para realizar todo lo relacionado con eventos, validación en el lado de servidor, conversión de datos, paginación, … y también de diferentes librerías que permiten añadir elementos a las páginas web así como relacionar los componentes de las vistas con los objetos java de la capa de negocio del servidor.
  El primer objetivo de utilizar la arquitectura que proporciona JSF es la separación entre la lógica de la aplicación y la presentación al usuario, teniendo a la misma vez la facilidad de conectar ambas capas sin la necesidad de utilizar ningún script. [41]

- Java Persistence API: Es un interfaz de programación de aplicaciones encuadrado en la especificación JSR 220 [43] que facilita al desarrollador de java el establecer una relación de mapeo de objetos con entradas de las tablas de las bases de datos con las que está trabajando la aplicación. A parte de realizar un mapeo de las bases de datos, también ofrece métodos basados en peticiones SQL (Structures Query Language) que permiten la interactuación entre de la aplicación y la base de datos. [41]

4.1.2 GlassFish

Como se ha especificado anteriormente, para poder utilizar la aplicación web que en este documento se especifica, se necesita un servidor que, entre otras características, sea compatible con la tecnología Java Enterprise. Con este objetivo

se ha escogido el servidor GlassFish que, además, se trata de un software de código abierto.

Anteriormente se mencionaba que la versión del servidor ha de estar en acuerdo con la versión de Java EE con la que se ha desarrollado el software en cuestión. Debido a esto, a pesar de que hay cuatro versiones de servidores GlassFish, la aplicación, únicamente puede funcionar con la última versión (GlassFish 4.0) [44] ya que solamente esa es compatible con Java EE 7.

4.1.3 Java DB

Java DB es el servidor encargado del manejo de la base de datos. Al igual que Java EE, pertenece a Oracle y se trata de una distribución de Apache Derby.

Su elección se ha basado principalmente en dos de sus características: la compatibilidad total que existe con la tecnología Java Persistence ya que utiliza Java Database Conectivity (JDBC), también de Oracle, que se trata de una API pensada para realizar conexiones y peticiones SQL con la base de datos y la otra razón es que se trata de un servidor de código abierto.

4.1.4 XHTML, JavaScript y CSS3

Como se ha dicho anteriormente, estas tres tecnologías se han implementado en la parte de "front-end". Todas ellas funcionan en conjunto en todas las vistas de la aplicación ya que son complementarias.

- *XHTML:* es un lenguaje muy parecido a HTML pero más robusto e idóneo para construir aplicaciones web. Al ser un lenguaje de marcas que está basado en la idea de XML, tiene que cumplir un esquema más estricto que HTML pero también se puede considerar una ventaja ya que los "parsers" son más sencillos de implementar.
  Esta tecnología está directamente asociada con la tecnología JSF y se ha de implementar en caso de que el "back-end" sea codificado con dicha tecnología.
- *JavaScript:* es un lenguaje de programación que es interpretado por el navegador directamente y que sirve para la interactuación de la aplicación con el usuario sin tener que enviar peticiones HTTP al servidor.
- *CSS3:* es la última versión de las hojas de estilos en cascada. Permite definir, mediante una serie de reglas, el estilo que se le quiere dar a la vista y que será interpretado por el navegador para mostrárselo al usuario.

4.2 Aplicación

En este apartado se explican los aspectos y características generales de la aplicación desarrollada, para un correcto y profundo entendimiento del funcionamiento de la misma.

4.2.1 Configuración inicial para Windows

Anteriormente se ha explicado que para el funcionamiento de la aplicación web, es necesario el uso de dos servidores. Uno de los servidores controlará la navegación por las páginas XHTML y el otro será el encargado de llevar todo lo relacionado con los accesos a la base de datos.

Ambos servidores están incluidos en el fichero comprimido que se ofrece en este pfg se encuentran: el servidor GlassFish 4.0 open source edition que es el encargado de gestionar las vistas de la aplicación y por otro lado el servidor Java DB de Oracle que es una distribución de Apache Derby y que es el que controlará los accesos a las tablas de la base de datos.

Una vez descomprimido el fichero, aparecerá una carpeta raíz llamada TSES con los archivos y carpetas que se pueden ver en la Figura 19.

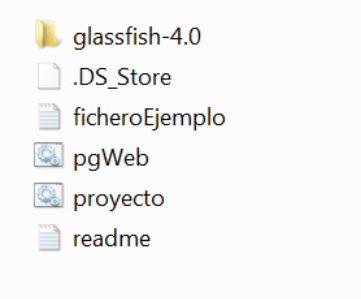


Figura 19. Carpeta TSES

Posteriormente, de debe leer el archivo “readme.txt” ya que en él se explica detalladamente todo lo necesario para poner en marcha la aplicación así como un listado con algunos de los problemas que se podrían ocasionar y sus posibles soluciones.

La información que aparece en el archivo es la siguiente:

//

Web application.

Author: Alejandro Rueda Pérez

ETSIST

//

PREPARACIÓN DEL ENTORNO.

Para su correcto funcionamiento comprobar que:

1. Está incluido la última versión de java en la variable del sistema (mínimo jre 7 y jdk 1.7.0)

2. En el archivo "java.policy" que se encuentra en la ruta: "Java\jre7\lib\security" se debe añadir el permiso:

Permission java.net.SocketPermission "localhost:1527", "listen";

En caso de que no estuviera anteriormente para permitir al servidor de base de datos escuchar en esa dirección.

Ejecute el archivo por lotes "proyecto".

INSTRUCCIONES DE ARRANQUE:

Para iniciar, siga los siguientes pasos:

1. Introduzca "start-domain" y espere a que finalice.
2. Introduzca "start-database" y espere a que finalice.

Una vez realizado, sin cerrar la ventana, ejecute el archivo por lotes "pgWeb".

Instrucciones de cierre:

Para finalizar los servidores siga los siguientes pasos:

1. Introduzca"stop-database" y espere a que finalice.
2. Introduzca "stop-domain" y espere a que finalice.
3. Introduzca “exit”.

//

TROUBLESHOOTING

1. Al hacer doble click aparece la pantalla de cmd pero se oculta de nuevo.

   --> Intente hacer click derecho e "Ejecutar como Administrador"

2. Se ha podido ejecutar pero la consola dice "java no se reconoce como un

   comando interno o externo".

   --> Compruebe que el path de java está correctamente referenciado en las variables del sistema.

Como se puede observar, la información del fichero se divide en tres grandes bloques: En el Primero se describen los requisitos iniciales (que básicamente son los que permitirán iniciar los dos servidores), después los procedimientos para el arranque del sistema y su puesta en marcha y finalmente los problemas que podrían ocurrir y cómo intentar solucionarlos, aunque hay que tener en cuenta que las soluciones que se ofrecen son de carácter general y que alguna vez no será suficiente para solventar el problema por lo que entonces habría que buscar una solución particular.

4.2.1.1 Preparación del entorno

Este paso es un requisito necesario ya que sin realizar todos los procedimientos indicados en el fichero, puede que el programa funcione adecuadamente. Esta sección está compuesta de dos pasos que son:

1. Comprobar que está instalado el JRE 7 y el JDK 1.7.0 como mínimo, aunque es recomendable que sea la última versión de Java que se pueda descargar de la página oficial de Oracle.
2. Proveer a la aplicación de un permiso especial para que el servidor de la base de datos pueda escuchar en el puerto que tiene asignado por defecto. Esto es necesario debido a la característica de Java de tener algunos ficheros de configuración que están por defecto preparados para proteger, sobre todo, los accesos a servidores que estén utilizando esta tecnología.

Para ello hay que buscar el archivo "java.policy" que se encuentra en la ruta relativa: "Java\jre7\lib\security". Un fragmento de la información que debe aparecer en dicho fichero es la siguiente:

```
// Standard extensions get all permissions by default

grant codeBase "file:${{java.ext.dirs}}/*" {
        permission java.security.AllPermission;
};

// default permissions granted to all domains

grant {
        permission java.lang.RuntimePermission "stopThread";
        // allows anyone to listen on dynamic ports
        permission java.net.SocketPermission "localhost:0", "listen";
        permission java.net.SocketPermission "localhost:1527", "listen";
```

Una vez abierto, se debe añadir, como se observa en el fragmento del fichero, la sentencia:

```
Permission java.net.SocketPermission "localhost:1527","listen";
```

Es posible que si se intenta modificar el fichero directamente en la carpeta que lo contiene no deje guardar los cambios por falta de derechos. En este caso habría que seguir el siguiente procedimiento:

1. Se copia el fichero en el escritorio
2. Se modifica con la nueva información y se guarda
3. Se vuelve a meter en la carpeta que lo contenía y cuando Windows pregunte qué hacer con el otro fichero se hace click en reemplazar.

4.2.1.2 Instrucciones de arranque

Se describen los procedimientos para, una vez instalado y puesto a punto todo lo necesario, iniciar los servidores para que escuchen a las peticiones de los usuarios en sus puertos determinados. Los pasos son:

1. Hacer doble click en el archivo ejecutable por lotes llamado "proyecto" obteniendo, si todo ha ido bien, la pantalla de la consola de Windows que se muestra en la Figura 20.

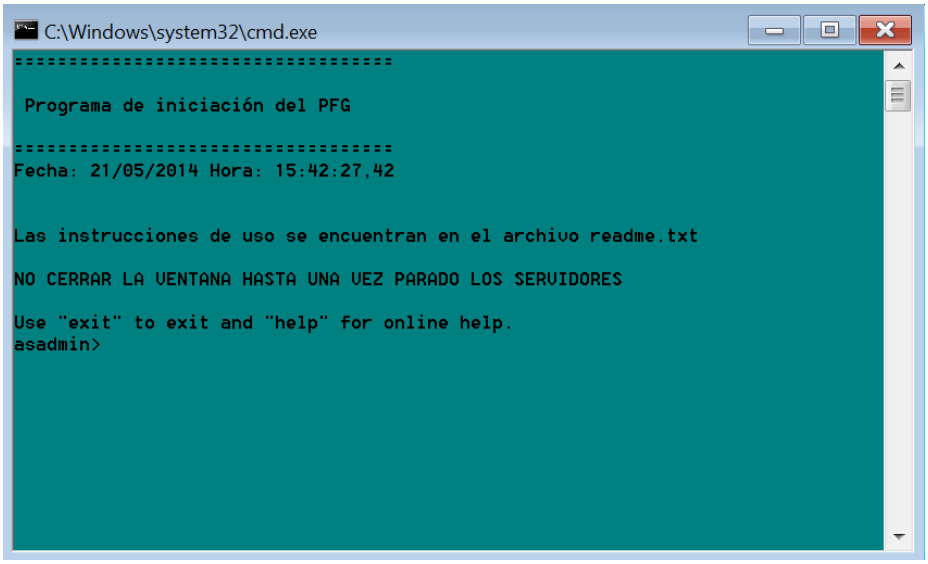


Figura 20. Consola Windows

2. Una vez abierta la pantalla de la Figura 20, se debe introducir el comando: "start-domain" que será el encargado de iniciar el servidor GlassFish 4.0 y que estará escuchando los puertos 8080 si se trata del protocolo http o 8181 si se trata del el protocolo https. El resultado de una correcta ejecución se muestra en la Figura 21.

```
asadmin> start-domain
Waiting for domain1 to start .......
Successfully started the domain : domain1
domain  Location: C:\Users\Alex\Desktop\Proyecto-PFG\glassfish-4.0\glassfish\dom
ains\domain1
Log File: C:\Users\Alex\Desktop\Proyecto-PFG\glassfish-4.0\glassfish\domains\dom
ain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
asadmin>
```

Figura 21. Start-domain

Como se puede observar en la Figura 21, se inicia el dominio 1 del servidor por defecto en el que va a correr la aplicación. También se ofrece la información del puerto donde se sitúa la consola de configuración del servidor que en este caso resulta ser en el puerto 4848.

3. Una vez finalizada la acción se permitirá la introducción de otro comando que, en este caso, es: "start-database" y que como su propio nombre indica, iniciará el servidor Java DB escuchando en el puerto 1527.
   Tras una correcta ejecución se mostrará por último la información que se puede ver en la Figura 22.

```
Starting database in the background.
Log redirected to C:\Users\Alex\Desktop\Proyecto-PFG\glassfish-4.0\glassfish\dat
abases\derby.log.
Command start-database executed successfully.
asadmin>
```

Figura 22. Start-database

4. Finalmente, se hará doble click en el archivo ejecutable por lotes "pgWeb" que abrirá la aplicación en el navegador que tenga el usuario seleccionado por defecto.

Para parar los servidores se ha de realizar el proceso inverso. En el caso de que se haya cerrado la ventana de la consola, se ha de hacer doble click en el archivo por lotes "proyecto" y seguir los siguientes pasos:

1. Introducir el comando "stop-database" que su correcta ejecución dará por salida la información que aparece en la Figura 23.

```
asadmin> stop-database
Wed May 21 16:46:28 FET 2014 : Se obtuvo la conexi≤n con el sistema principal:
.0.0.0, n·mero de puerto 1527.
Wed May 21 16:46:28 FET 2014 : Apache Derby Network Server - 10.9.1.0 - (134487
) cierre
Command stop-database executed successfully.
asadmin>
```

Figura 23. Stop-database

2. Introducir el comando "stop-domain" que parará el servidor GlassFish como se indica en la Figura 24.

```
asadmin> stop-domain
Waiting for the domain to stop .
Command stop-domain executed successfully.
asadmin>
```

Figura 24. Stop-domain

3. Por último, introducir el comando "exit".

4.2.1.3 Troubleshooting

Como se puede ver en la información que presenta el documento, solamente se han incluido dos posibles fallos que puedan ser resueltos por el usuario y que son los más comunes que pueden ocurrir en la fase de instalación del sistema.

1. Puede ser que, al tratarse de un archivo por lotes, un usuario que no tenga derechos de administrador tenga problemas para poder ejecutarlo por cuestiones de seguridad por parte del sistema operativo. Para ello se ha de hacer click con el botón derecho en el archivo y elegir la opción de "Ejecutar como administrador".
2. El segundo error que aparece se produce cuando el entorno de Java no está instalado correctamente en el sistema o no se ha referenciado correctamente en las variables del entorno del sistema.

4.2.2 Configuración inicial para sistemas basados en Unix

Desde la misma carpeta donde se aloja la aplicación, existe también la posibilidad de iniciar los servidores para que estén funcionando en un sistema operativo basado en Unix.

El procedimiento es el mismo, se ha de seguir los mismos pasos indicados anteriormente excepto por las siguientes dos diferencias:

1. El archivo que se debe iniciar al principio para introducir los comandos correspondientes es un script ejecutable de bash que, al igual que para el sistema operativo Windows, se encuentra en la raíz de la carpeta principal con el nombre "proyecto".
2. Cuando se introduce el comando "start-database" hay que tener en cuenta que el servidor de bases de datos, que en este caso es "JavaDB" fijará por defecto la ruta donde se deben de encontrar las bases de datos con las que se va a trabajar.
   La ruta por defecto para este servidor es: "/Users/{NombreDeUsuario}" donde "{NombreDeUsuario}" como su propio nombre indica, hace referencia al nombre de sesión de usuario del sistema operativo en el que se quiere iniciar los servidores.
   Consecuentemente, en esa dirección es donde se debe copiar la carpeta de nombre: "InternetOfThings" que se encuentra en: "./Proyecto-PFG/glassfish-4.0/glassfish/" como se observa en la Figura 25, en la carpeta de la ruta por defecto del servidor de bases de datos.

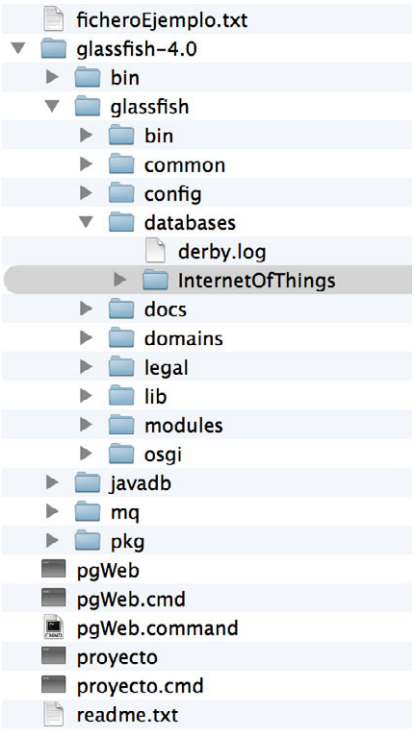


Figura 25. Carpeta "InternetOfThings"

4.2.3 Seguridad

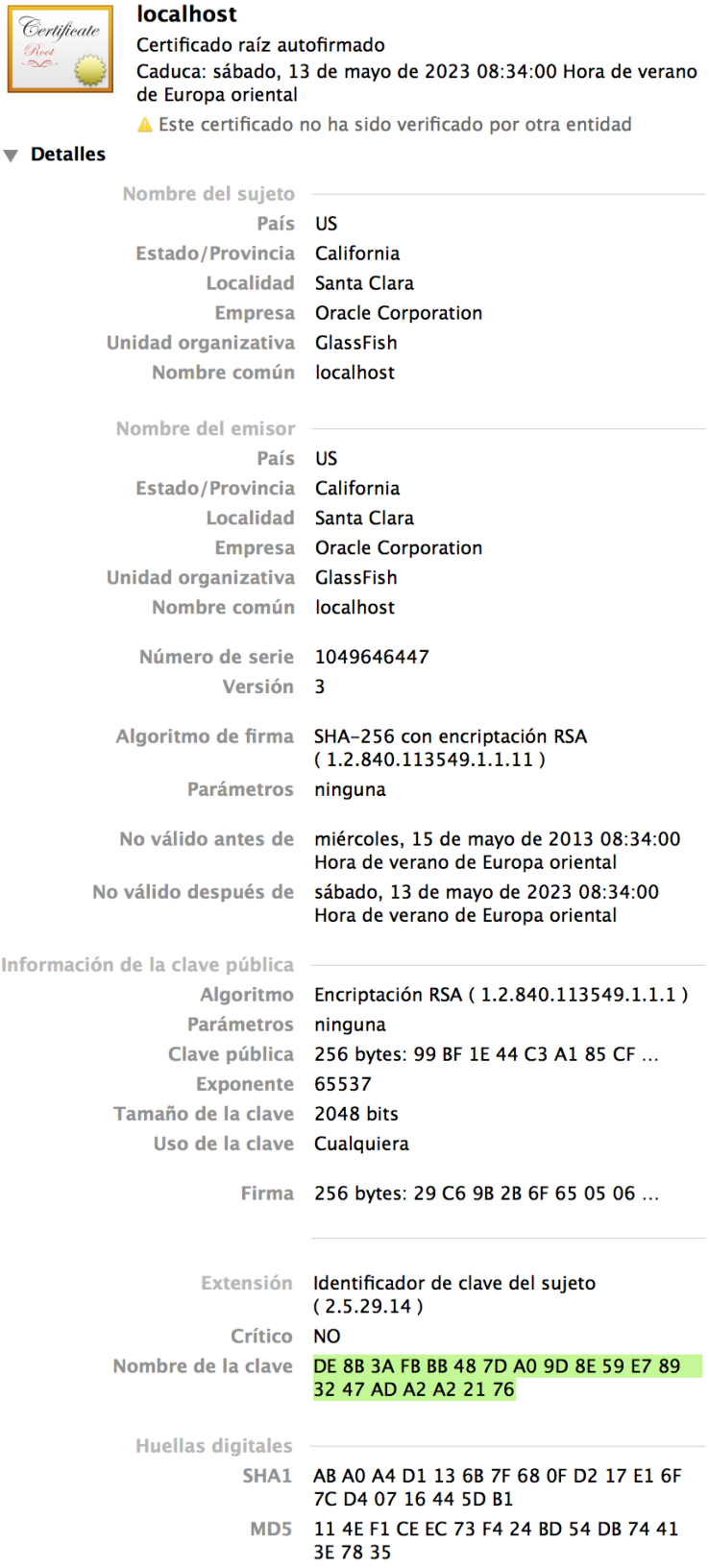
En la aplicación web se han implementado dos medidas de seguridad, por una parte la comunicación con el servidor se realiza a través del protocolo de aplicación Hipertext Transfer Protocol Secure, y por otra parte se han establecido dos grupos de usuarios dentro de la aplicación que son: administradores o expertos en seguridad y usuarios comunes.

4.2.3.1 HTTPS

Como se ha mencionado anteriormente, para todas y cada una de las conexiones con la aplicación web, se utiliza el protocolo de seguridad HTTPS que es un protocolo de aplicación que tiene como objetivo cifrar el canal de transmisión y que se basa en SSL/TLS (Secure Sockets Layer/Transport Layer Security).

Esta característica, es introducida directamente por el servidor ya que se configura en el descriptor de la aplicación y en éste caso se ha decidido que en todas las vistas del programa se utilice dicho protocolo de seguridad.

En el caso particular de la versión que se ha utilizado de GlassFish (4.0), el certificado que se provee es el de la Figura 26:



localhost

Certificado raíz autofirmado

Caduca: sábado, 13 de mayo de 2023 08:34:00 Hora de verano de Europa oriental

⚠ Este certificado no ha sido verificado por otra entidad

▼ **Detalles**

Nombre del sujeto	
País	US
Estado/Provincia	California
Localidad	Santa Clara
Empresa	Oracle Corporation
Unidad organizativa	GlassFish
Nombre común	localhost
Nombre del emisor	
País	US
Estado/Provincia	California
Localidad	Santa Clara
Empresa	Oracle Corporation
Unidad organizativa	GlassFish
Nombre común	localhost
Número de serie	1049646447
Versión	3
Algoritmo de firma	SHA-256 con encriptación RSA (1.2.840.113549.1.1.11)
Parámetros	ninguna
No válido antes de	miércoles, 15 de mayo de 2013 08:34:00 Hora de verano de Europa oriental
No válido después de	sábado, 13 de mayo de 2023 08:34:00 Hora de verano de Europa oriental
Información de la clave pública	
Algoritmo	Encriptación RSA (1.2.840.113549.1.1.1)
Parámetros	ninguna
Clave pública	256 bytes: 99 BF 1E 44 C3 A1 85 CF …
Exponente	65537
Tamaño de la clave	2048 bits
Uso de la clave	Cualquiera
Firma	256 bytes: 29 C6 9B 2B 6F 65 05 06 …
Extensión	Identificador de clave del sujeto (2.5.29.14)
Crítico	NO
Nombre de la clave	DE 8B 3A FB BB 48 7D A0 9D 8E 59 E7 89 32 47 AD A2 A2 21 76
Huellas digitales	
SHA1	AB A0 A4 D1 13 6B 7F 68 0F D2 17 E1 6F 7C D4 07 16 44 5D B1
MD5	11 4E F1 CE EC 73 F4 24 BD 54 DB 74 41 3E 78 35

Figura 26. Certificado de seguridad

Es un certificado que no está reconocido por defecto en la mayoría de los navegadores ya que es propio del GlassFish de Oracle. Por tanto es muy probable

que al intentar ejecutar por primera vez se muestre un mensaje parecido al de la Figura 27.

Figura 27. Mensaje de confianza en el navegador

Para acceder finalmente a la aplicación, se haría click en "Continuar de todos modos".

4.2.3.2 Roles de seguridad

Desde el punto de vista del usuario, se pueden distinguir dos grupos en la aplicación que van a tener diferentes roles que a su vez determinarán qué acciones pueden realizar dentro del sistema.

- **Administradores del sistema:** También llamados expertos de seguridad, son aquellos encargados de añadir, borrar y modificar cualquiera de los elementos de las tablas de la base de conocimientos, es decir, manejar los datos relacionados con los ataques, mecanismos, redes y servicios de la base de conocimientos.
  Estos expertos deben de tener el conocimiento suficiente acerca de la base de conocimientos de tal forma que al modificar la misma no se pueda introducir alguna incongruencia ya que los datos de esta base de conocimientos son muy sensibles debido a su utilización en la lógica del programa y se podría, por lo tanto, causar un mal funcionamiento dentro de la misma.

  Por otro lado, los administradores también tienen la posibilidad de chequear el log del programa y con ello ver los posibles fallos en caso de que no funcione correctamente alguna de las características ofrecidas por el mismo o que se haya producido algún resultado erróneo. Por tanto, se podrá revisar de forma exhaustiva los procedimientos que, paso a paso, han llevado al programa a producir una determinada salida.

  Por último, se hace necesario mencionar que un administrador tiene también todas las posibilidades que se le ofrece al otro grupo que es el de usuario común.

- **Usuarios:** Tienen todas las posibilidades de actuación excepto la de modificar datos en la base de conocimientos y la de revisar el log que son exclusivas de los administradores.
  El objetivo de este tipo de usuario es que cualquier persona que no sea experto pero desee utilizar el programa pueda, mediante la introducción de ciertos imperativos, determinar cuál serían sus opciones siempre teniendo en cuenta únicamente los datos existentes en la base de conocimientos. Por esto último, también se ofrece la posibilidad al usuario de explorar los datos que hay en todo momento en la base de conocimientos.

Finalmente, cabe mencionar que el sistema de seguridad por roles se ha establecido en torno a un exhaustivo control de sesión llevado a cabo por el propio servidor. Debido a esto, en el mismo momento que finaliza la sesión, el usuario ha de autenticarse de nuevo en el caso de que desee continuar utilizando la aplicación web.

Existen dos maneras de terminar la sesión que se está llevando a cabo:

- **Finalización activa**: es cuando el usuario decide acabar con su sesión y hace click en el botón de desconexión de alguna de las vistas como se explicará posteriormente.
- **Finalización pasiva**: con el objetivo de controlar la sesión, el servidor, (en este caso el GlassFish 4.0) dispone de un contador del que se puede variar el tiempo y que vigila que el usuario de esa sesión todavía está activo y que cierra la misma cuando ya no lo está. Para esta aplicación el tiempo de sesión es de 30 segundos por lo que, por ejemplo, en el caso de que un usuario se haya autenticado con anterioridad y en un determinado momento deje de hacer peticiones durante 30 segundos, el servidor supondrá que esa sesión ha terminado y la cerrará automáticamente obligando a dicho usuario a poner de nuevo sus credenciales en caso de que quiera volver a hacer alguna petición al servidor y éste a su vez le asignará una nueva clave de sesión.

Para poder añadir, borrar o modificar usuarios dentro del sistema, es necesario acceder a la consola del servidor y ahí, dentro del realm de seguridad[4] que se esté utilizando, realizar las acciones convenientes[5].

Por último, mencionar que dentro de la aplicación, existen tres vistas relacionadas con los procesos de autenticación: el log in, el error de autenticación y el error de seguridad.[6]

[4] Un realm de seguridad es un conjunto o ámbito donde se recogen políticas de seguridad para un determinado grupo de usuarios.
[5] Ver anexo documento "Manual de usuario"

4.2.4 Base de conocimientos

La base de conocimiento es, quizás, la parte más importante de la aplicación web junto con el algoritmo de la lógica central del programa. Se entiende totalmente necesario el hecho de que la base de conocimientos esté debidamente alimentada ya que así el programa devolverá una salida mejor y más precisa.

La base de conocimientos es, en realidad, una serie de tablas situadas en una base de datos que utilizará la lógica de programa para realizar las funciones necesarias y así ofrecer un resultado coherente de salida. Con este objetivo se ha decidido realizar cuatro tablas: ataques, mecanismos, servicios y redes.

Ataques

Incluye cada uno de los ataques conocidos que afectan a una o más redes de la tabla de redes. Como se observa en la tabla 1, aparecen los campos:

- *Nombre:* Es el nombre que se le ha dado al ataque que tiene que ser a su vez único.
- *Servicios de seguridad:* Son los servicios específicos de seguridad a los que dicho ataque puede afectar.
- *Mecanismos de seguridad:* Son los mecanismos que se podrían implementar para paliar dicho ataque.
- *Tipo de red:* Es el tipo de red al que puede afectar el ataque en cuestión.
- Comentarios

Nombre	Servicios de seguridad	Mecanismos de seguridad	Comentarios	Tipo de red
Ataque 1	Servicio 1, Servicio2	Mecanismo 1	*Referencia*	Red 1, Red 3
…	…	…	…	…
Ataque n	Servicio 3	Mecanismo 3, Mecanismo 5	*Referencia*	Red 1

Tabla 1. Ataques de la base de conocimientos

Servicios de seguridad

Existen seis principales grupos de servicios de seguridad que se han denominado en el presente documento como servicios generales y son los siguientes:

- Autenticación
- Control de acceso
- Confidencialidad
- Integridad

[6] Ver anexo documento "Manual de usuario"

- No repudio
- Disponibilidad

A cada servicio general, le puede corresponder uno o más servicios específicos. En la Tabla 2 se puede ver cómo se ha dispuesto esta relación en la base de conocimientos.

Servicio	**Servicio específico**
Servicio 1	Servicio específico 1
Servicio 1	Servicio específico 2
Servicio 2	Servicio específico 3
Servicio 2	Servicio específico 4

Tabla 2. Servicios de la base de conocimientos

Mecanismos

Son los mecanismos que pueden paliar uno o más ataques de los que aparecen en la tabla de ataques. Sus características, recogidas en la tabla 3, son las siguientes:

- *Nombre:* Es el nombre del mecanismo y que debe ser único.
- *Capas:* Son las capas en las que se puede implementar el mecanismo en cuestión.
- *Tipo de red:* Es el tipo de red en el que se puede implementar el mecanismo.
- *Comentarios*

Nombre	**Phy**	**Link**	**Net**	**Trans**	**App**	**Tipo de Red**	**Comentarios**
Mecanismo A	x	x				Red 1	
Mecanismo B			x			Red 1, Red 2	
…							
Mecanismo N		x	x			Red 5	

Tabla 3. Mecanismos de la base de conocimientos

Redes

Son las características que puede tener una red y que va a diferenciar una red de otra. Esto es importante porque existen multitud de tipos de redes y como se ha explicado anteriormente, sobre todo en las WSN hay varios factores relacionados con la estructura, arquitectura y componentes de la misma que influyen directamente en los mecanismos y servicios de seguridad que se pueden aplicar a dicha red y por tanto a los resultados de salida que ofrecerá el algoritmo del software de la aplicación web.

Nombre	Recursos	Conectividad	Comunicación	Recursos en estación base	Protección	Topología
Red 1	Batería Memoria	Enlace Aire	Envío-Respuesta	Proceso	Manejo de fallos	Malla
…						
Red n						

Nombre	Nodos	Encaminamiento	Señalización	Sincronismo	Estación base
Red 1	Agregación, Tránsito	x	x	x	Conexión sink-nodo, Proc. de resultados, Registro y proc. información
…					
Red n					

Tabla 4. Redes de la base de conocimientos

Como se observa en la tabla 4, para una red se pueden especificar los siguientes campos:

- *Nombre:* Un nombre específico que se le quiere dar a la red y éste debe de ser único en la base de datos.
- *Recursos:* En los nodos de la red pueden haber ciertas limitaciones de recursos como puede ser de batería, memoria, procesamiento, alerta local lumínica y alerta local sonora.
- *Conectividad:* Es el tipo de conexión entre nodos y puede ser tanto por cable como por Wireless.
- *Comunicación:* Se puede tratar de una comunicación en modo "petición-respuesta" o broadcast.

- *Recursos en estación base:* Al igual que ocurre en los nodos, la estación base también puede tener ciertas limitaciones, que además, coinciden con las de los nodos.
- *Protección:* Si cuenta con algún tipo de protección como puede ser de manejo de fallos, modo autónomo o de estanqueidad.
- *Topología:* Que puede ser de estrella, malla o mixta.
- *Nodos:* Se indican los tipos de nodos que se incluyen en la red. Las posibilidades que se ofrecen son: de soporte, tránsito, o agregación.
- *Encaminamiento:* Se indicará si existe encaminamiento en la red.
- *Señalización:* Para especificar si hay señalización.
- *Sincronismo:* Sirve para indicar si hay o no sincronismo.
- *Estación base:* Describe brevemente las características de la estación base. Se ofrecen las posibilidades de: conexión sink-nodo, recepción y procesado de agregados, procesamiento de resultados y registro y procesamiento de información.

4.2.5 Funcionamiento lógico del sistema

La lógica del programa es la parte más importante a la hora de obtener a la salida unos resultados coherentes. Una vez realizado el análisis correcto de la información de entrada y la base de conocimiento que deberá estar debidamente completada, se han de desencadenar diferentes procedimientos que determinarán un tipo de salida u otro.

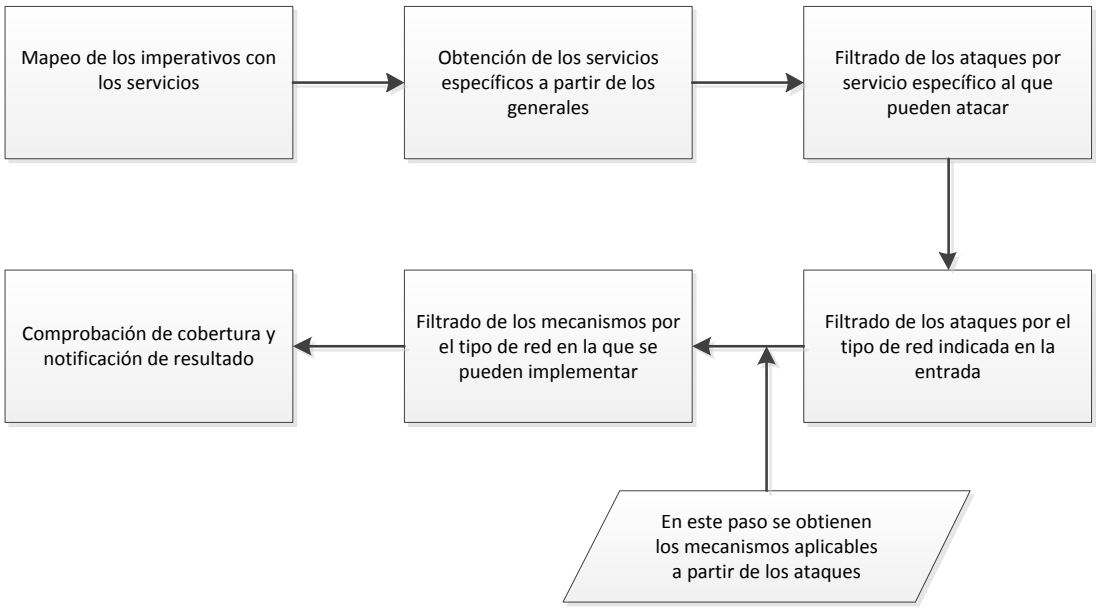


Figura 28. Lógica principal de la aplicación

Para una mayor comprensión de la lógica principal, se ha considerado realizar un seguimiento exhaustivo a través de la misma con unos datos que podrían ser procedentes de un escenario de actuación real.

Por tanto, partiremos del supuesto de que existan un total de 50 ataques en la Tabla de Ataques de la base de conocimientos.

Los servicios de seguridad que se podrían aplicar, se obtienen mediante el mapeo directo de los imperativos legales de la entrada. Una vez obtenidos los servicios generales, se obtiene de la Tabla de Servicios (Tabla 2) los servicios específicos de seguridad para cada uno de los servicios generales que se han mapeado anteriormente.

Una vez obtenidos los servicios específicos que se quieren proveer, para cada uno de ellos, la lógica filtra el número total de ataques (50 en este caso) y confecciona una lista con todos aquellos que afecten al menos a uno de los servicios específicos.

Supongamos que en este ejemplo fueran 30 los ataques que quedaran después del primer filtro ya que serían los que afectarían al menos a un servicio específico.

Estos 30 ataques afectan a los servicios de seguridad indicados en la entrada al sistema pero no todos los ataques afectarían al tipo de red que se quiere implementar, por lo que habría que filtrar de nuevo comprobando cuáles de dichos ataques pueden ser una amenaza para ese determinado tipo de red. En el ejemplo de desecharían 10 de los 30 ataques resultantes del anterior filtro.

Cada uno de esos ataques tiene la siguiente estructura con relación a los mecanismos:

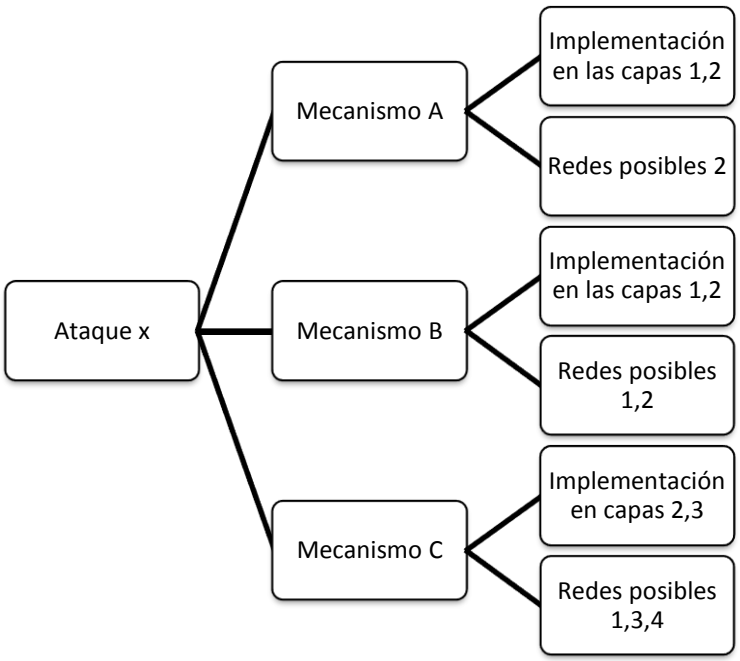


Figura 29. Estructura de relación ataques-mecanismos

Para cada uno de los 20 ataques se ha de realizar un filtro de mecanismos, es decir, un ataque puede tener varios mecanismos para mitigarlo pero no todos los mecanismos se pueden aplicar a todas las redes por lo que el siguiente paso sería basándose en el tipo de red de la que se está tratando, ver qué mecanismos se pueden finalmente implementar para mitigar el ataque en concreto. Esto se realizaría gracias a la Tabla 3.

De la misma forma se ha de ver que existen mecanismos eficientes para todos los ataques que se han encontrado en el paso anterior y que puede haber una cobertura completa en la red. Puede ocurrir que para un determinado ataque no exista un mecanismo que se pueda implementar para el tipo de red indicado. En este caso se tendría que notificar en la salida para que el usuario pueda o bien, cambiar el tipo de red o los imperativos legales de la entrada.

Por ejemplo, la lógica del programa para el caso del Ataque X reflejado en la Figura 29, considerando que nuestra red fuera:

- **Red de tipo 1:** La lógica del programa sabe que tiene que buscar mecanismos que se puedan implementar en la red de tipo 1 y de esos tres mecanismos podrían valer los dos últimos ya que el primero solo valdría para redes de tipo 2. Posteriormente se notifica las capas de implementación de cada mecanismo.
- **Red de tipo 4:** La lógica del programa detectaría que la única posibilidad sería aplicar el mecanismo 3 por el filtro de tipo de red. Posteriormente se notifica las capas de implementación de cada mecanismo.
- **Red de tipo 7:** La lógica del programa debería notificar que no existen mecanismos que se puedan aplicar a esa red para el ataque x. Por lo que el ataque x no quedaría cubierto para ese tipo de red y sin embargo podría afectar a esa red.

Finalmente se notificaría al usuario tanto si existe o no una cobertura completa para los parámetros de entrada introducidos y si es así, se expondrían los posibles servicios de seguridad y mecanismos de seguridad que se necesitarían implementar en dicha red.

4.2.6 Informes de cobertura de conocimientos

La aplicación dispone de un sistema de detección de fallos consistente en un archivo en el que se va guardando toda la información correspondiente a cualquier acción realizada tanto por parte del usuario como interna del sistema. Todas y cada una de las veces que se guarda alguna de estas entradas, junto con ella, irá también con su time-stamp correspondiente.

El objetivo principal de esto es que en caso de que se detecte que la información de salida es incongruente, se pueda visualizar cada una de los procesos internos que se han llevado a cabo y que no se muestran a un usuario normal y así tener la posibilidad de detectar en qué momento se produce el fallo de concepto que lleva a

una salida incorrecta. Principalmente, lo que este método proporciona, es la ventaja de realizar un análisis de los resultados por parte de un experto de seguridad y así poder prevenir futuros resultados erróneos por parte de la aplicación.

El usuario con rol de administrador, podrá acceder en todo momento desde el menú principal a una vista donde se muestra la información de dicho archivo en incluso se da la posibilidad de descargarlo con formato “txt..

El archivo se puede encontrar con el nombre de “log.txt” en la ruta:

```
./glassfish-4.0/glassfish/domains/domain1/config
```

4.3 Información de entrada

Como se ha explicado con anterioridad, la entrada del sistema consta de una serie de imperativos que influirán directamente en el resultado de salida del programa en función también del tipo de red introducido y del estado de la base de conocimientos de la aplicación en ese determinado momento.

Los imperativos que se puede recibir a la entrada son los reflejados en la tabla 1 junto con los datos a los que se aplican.

Security imperatives	***Applied data***
Truthfulness of the actors	Nodes´ ID Connections´ ID Users´ ID Physical access
Access authorization	Users Nodes Connections
Transactions´authorization	Users Systems Nodes
Disclosure	Collected data Identities data Routing data

Security imperatives	***Applied data***
	Exchanged data Aggregated data Disaggregated data Activity trace User´s data Statistical data Buffers Servers´ data
Content´s veracity	Collected data Identities data Routing data Exchanged data Aggregated data Disaggregated data Activity trace User´s data Statistical data Buffers Servers´ data
Actors´accountability	
Availability	

Tabla 5. Imperativos y datos a los que se aplican

Los tipos de redes que se pueden tratar en la entrada son exclusivamente aquellas que se hayan guardado con anterioridad en la base de conocimientos. Para escoger el tipo de red que se quiere introducir en la lógica, se debe de especificar el nombre completo de la red tal y como aparece en el campo "Nombre" de la Tabla de redes (Tabla 4).

4.3.1 Fichero

Una de las formas de proceder a una evaluación inmediata en el software es a través de un fichero que, con un determinado formato, se puede subir al servidor para ser analizado y así evitar la introducción manual de los datos de entrada del programa por el usuario.

En un principio está pensado para ser una entrada automática procedente del subsistema anterior LES en la que los datos vendrían estructurados siguiendo un patrón lógico que, posteriormente, pudiera ser interpretado por este subsistema para su correcto análisis.

En el caso de que la entrada del subsistema fuera la información de salida del subsistema LES, no se entendería como necesario contemplar la estructura de dicho fichero ya que saldría correctamente estructurado del funcionamiento lógico del subsistema LES y por tanto, no podrían producirse errores "humanos" en el momento de la creación del fichero.

Puesto que, como se ha indicado con anterioridad, el fichero va a ser en un principio manejado por usuarios humanos, se ha decidido que el formato más adecuado es el "txt" (formato de texto plano). El principal motivo de esta decisión es que este formato es el que permite su total comprensión con una simple lectura, así como de su redacción. Otra de las razones que ha motivado a elegir este formato de fichero es que prácticamente cualquier persona puede conseguir hacer un fichero que pueda entender el programa y por tanto puede ser usado por una mayor cantidad de personas, aparte de que prácticamente cualquier ordenador del mundo, tiene un editor de texto plano.

Una vez expuesta la motivación del formato elegido se procederá a explicar la manera de construir un documento que pueda ser correctamente interpretado por el programa. La importancia de seguir una buena estructura es debido a que, con el objetivo de la previsión de fallos en la lógica principal, se ha introducido un mecanismo de validación del fichero por el cual, solamente, funcionará si la información que hay dentro del mismo pasa el proceso de validación.

A continuación se muestra un ejemplo de fichero con la estructura adecuada:

```
%Esto sería un comentario del fichero

Red 3

*
Veracidad de los actores
ID Nodos
```

ID Usuarios
ID Conexiones

*

Autorización de acceso
Usuarios
Nodos
Conexiones

*
Revelación
Datos recopilados
Datos de routing

*
Veracidad del contenido
Datos recopilados
Datos identificativos
Datos de routing

*
Rendición de cuentas de los actores

*
Continuidad del servicio

Antes de nada, es importante saber que las líneas en blanco no influyen en nada dentro del documento.

Como se observa en la primera línea, los comentarios dentro del fichero van precedidos por un símbolo de tanto por ciento (%). Éstos, como su nombre indica, no serán procesados ni por el validador ni por la lógica del programa.

Lo primero que va a buscar la lógica, es el nombre de la red con la que se está tratando. Aquí es donde empieza a actuar el validador ya que va a comprobar que la

red introducida esté dentro de la base de datos y en caso de que no esté en la base de conocimientos, sacará por pantalla un mensaje del tipo:

Incorrect information in the document - Network type is not in the database

Después vienen los bloques de Imperativos Legales. Como se observa, están precedidos de un asterisco (*) que debe de ir en una línea. En caso de que la lógica encuentre otra línea que sea texto inmediatamente después de haber leído el nombre de red, dará un error de validación diciendo:

Not-well formed document- Error in line (número de línea)

La línea siguiente se interpretará única y exclusivamente como el nombre del imperativo legal en cuestión y dicho imperativo tiene que ser uno de los que se muestran en la tabla 1 y debe de estar escrito exactamente igual o dará un mensaje de error de validación como el siguiente:

Incorrect information in the document- Error in line (número de línea)

Las líneas posteriores son los diferentes elementos a los que se le puede aplicar el imperativo en cuestión y pueden ser ninguno o varios. De momento no está determinado si se van a ir añadiendo más imperativos a la lógica en un futuro, por ello, todavía no se ha implementado un mecanismo de validación para la comprobación de dicho fenómeno. Tampoco se comprobará si la información es correcta.

En el apartado visual, para poder subir el fichero al servidor se dispone de una pantalla a la cual se accede directamente desde el menú principal de la aplicación.[7]

4.3.2 Formulario

Esta es la otra posibilidad de evaluación donde el usuario puede escoger dinámicamente tanto el tipo de red como qué imperativos y datos (a los que se aplican dichos imperativos) quiere introducir en la lógica del programa.

Para ello, desde el menú principal de la aplicación, se puede acceder a una vista[8] en la que se ofrece las opciones de red e imperativos correspondientes.

[7] Ver anexo documento “Manual de usuario”
[8] Ver anexo documento “Manual de usuario”

4.4 Resultados del sistema

Una vez explicadas las entradas del sistema y la lógica del programa, se hace necesario comentar cuál va a ser el formato de salida. Resulta una obviedad el mencionar que el contenido de dichas salidas va a depender directamente tanto de las entradas al sistema como del contenido de la base de conocimientos en el momento de la ejecución.

El formato de los resultados de salida del sistema está pensado para dos objetivos principales: el primero es que pueda ser visualizado y entendido por el usuario a través del interfaz de la aplicación y el segundo es que, como se ha comentado en el apartado cuatro, pueda ser entendido y analizado por el siguiente subsistema que es el subsistema de decisión.

Por tanto, los resultados que se ofrecen una vez procesada la entrada de este sistema son meras posibilidades de aplicación, de las cuales, el subsistema correspondiente elegirá cuál es la más apropiada en el caso particular de estudio.

Actualmente, la aplicación consta de dos vistas de resultados: base de hechos y los resultados. [9]

4.4.1 Base de hechos

En la base de hechos es donde se encuentra la información que, tras un análisis exhaustivo de la entrada, es la que se va a utilizar en la lógica del programa para poder así obtener resultados de salida sin errores.

Para ello, se procede como se indica en la Figura 30, obteniendo así toda la información que hay en la base de conocimientos acerca de la red que el usuario ha especificado en la entrada y también la información de los imperativos legales y los datos a los que se aplican esos imperativos.

[9] Ver anexo documento “Manual de usuario”

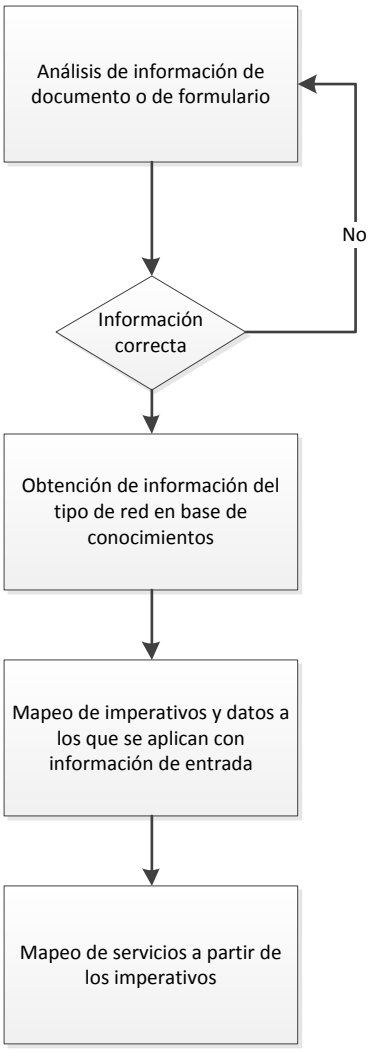


Figura 30. Lógica de obtención de la base de hechos

4.4.2 Resultados finales

Una vez confirmado que la información de la base de hechos es correcta, se procederá al análisis y cómputo de dicha información en la lógica principal del programa dando lugar a los resultados finales presentados en dos formatos: modo visual en pantalla y un documento de texto.

- Modo visual: Es el modo pensado para el usuario y se divide en tres partes. La primera parte es donde se indican los servicios generales que se han de implementar en la red introducida por el mismo. La segunda parte es donde aparecen todos los ataques que pueden afectar a la red y se indica si al menos existe un mecanismo para paliar cada uno de los ataques, lo que significaría que al menos existe una solución posible para tener una cobertura total para esa red con los imperativos indicados anteriormente. La tercera consiste en un resumen de todos los mecanismos aplicables en forma de tabla.

- Documento: Al igual que en formato visual, también se ofrece la posibilidad de descargar un documento en el que se ofrecerá información de: los servicios generales y específicos que se han de implementar en la red y también el segundo apartado del modo visual completo.

4.5 Caso práctico

4.5.1 Contexto inicial

Suponiendo, como estado inicial, una aplicación encuadrada en el entorno de la salud en la que hubiera que controlar el estado de personas mayores que se encuentran en sus domicilios para así, no obligar a esta fracción de población a tener que desplazarse a pasar consulta cada poco tiempo. Para este objetivo, es necesario implementar una red WSN en cada persona que se quiera monitorizar, que se encargará de recopilar datos sensibles relacionados con la salud de dicho anciano.

Ejemplos de estos datos serían la presión sanguínea, niveles de azúcar en sangre para los diabéticos, saturación de oxígeno, ritmo cardiaco, etc. Todas y cada uno de ellos tienen en común que son datos muy sensibles y que es necesario que gocen de una protección específica de acuerdo con la Ley Orgánica de Protección de Datos (LOPD) [45].

Una vez analizada la situación inicial, el sistema LES, teniendo en cuenta las características específicas del contexto de la aplicación y contrastándolas con la LOPD, dará el siguiente resultado de salida en forma de imperativos que corresponde con la entrada del sistema TSES.

Security imperatives	***Applied data***
Truthfulness of the actors	Nodes´ ID Physical access
Access authorization	Nodes
Disclosure	Identities data User´s data
Content´s veracity	Collected data
Availability	

Tabla 6. Imperativos de entrada del ejemplo de funcionamiento

Revelación: No se podrán revelar los datos de la identidad de los nodos dentro de la red ni de la identidad de la persona.

- ID de nodos
- ID personal

Disponibilidad: La monitorización debe de ser continua en todo momento.

Veracidad de los actores: Se debe de verificar que los nodos que están transmitiendo información son los nodos oficiales y no réplicas. También se necesita evitar el acceso físico a toda persona ajena al sistema.

- ID nodos
- Acceso físico

Autorización de acceso: Solo pueden acceder a la red los nodos autorizados.

- Nodo

Veracidad del contenido: Los datos recopilados en la estación base deben de ser exactamente los mismos que en su origen.

- Datos recopilados

También es importante mencionar que las salidas que se proporcionan en el caso práctico dependen también de la información que hay en la base de datos en el momento de la ejecución del mismo[10]

4.5.2 Ejecución de TSES

En este caso la red de entrada es una WSN con algunas restricciones en cuanto a los nodos y a la estación base, de topología de estrella y de conectividad de tipo broadcast.

Partiendo de la salida que ofrecerá el sistema LES en este caso práctico, se procederá a introducir los imperativos legales en la vista de "Evaluación rápida"[11] como se ilustra en la Figura 31.

[10] Ver anexo "Base de Conocimientos en el momento de las pruebas"
[11] Ver anexo "Manual de Usuario"

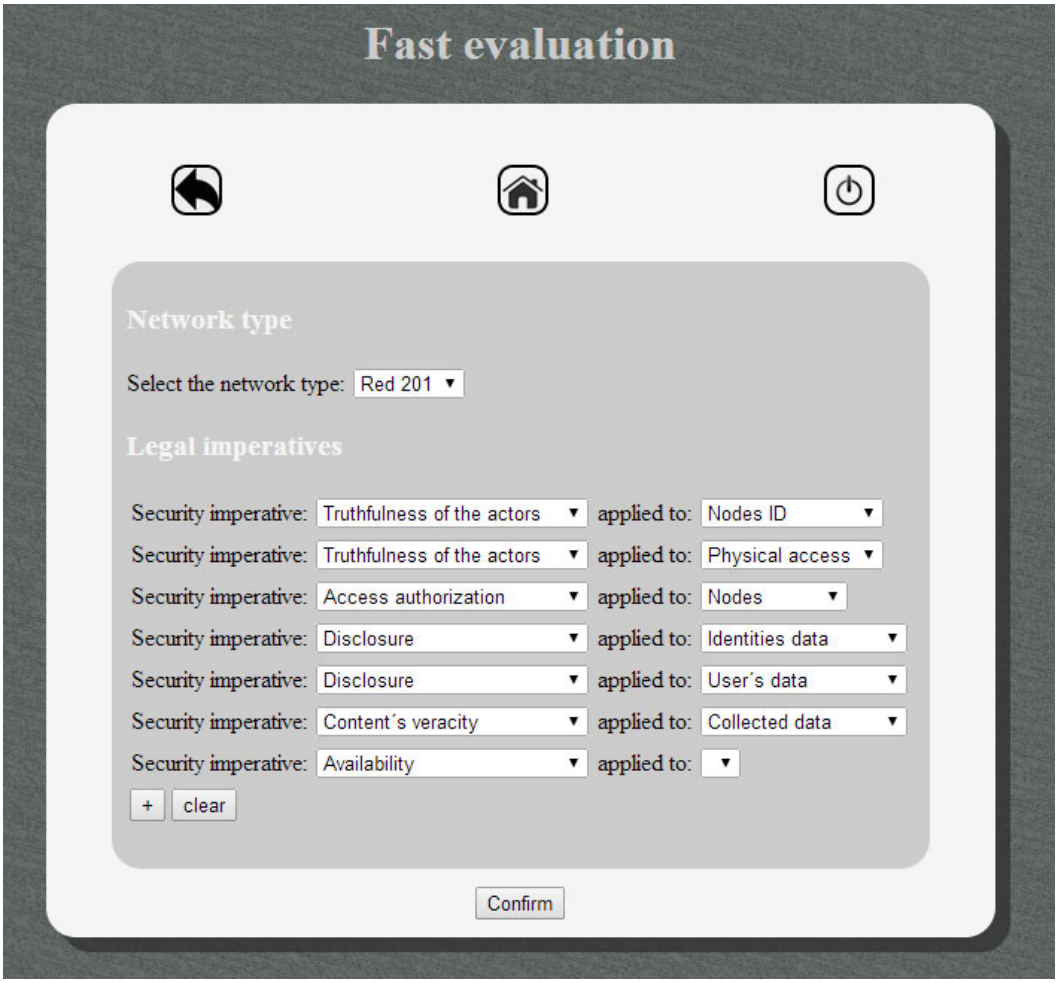


Figura 31. Imperativos de entrada en caso práctico

Se observa en la Figura 31 que la red sería del tipo “Red 201” (se ha considerado para los casos prácticos y de pruebas que las redes WSN son las del tipo “Red 2xx). También se puede comprobar que todos los imperativos con sus respectivos datos a los que se aplican han sido igualmente introducidos. Al confirmar aparece la base de hechos de la Figura 32 donde se muestran las características y limitaciones de la red “Red 201” así como a los imperativos interpretados y que son los que se van a tener en cuenta a la hora de obtener la salida.

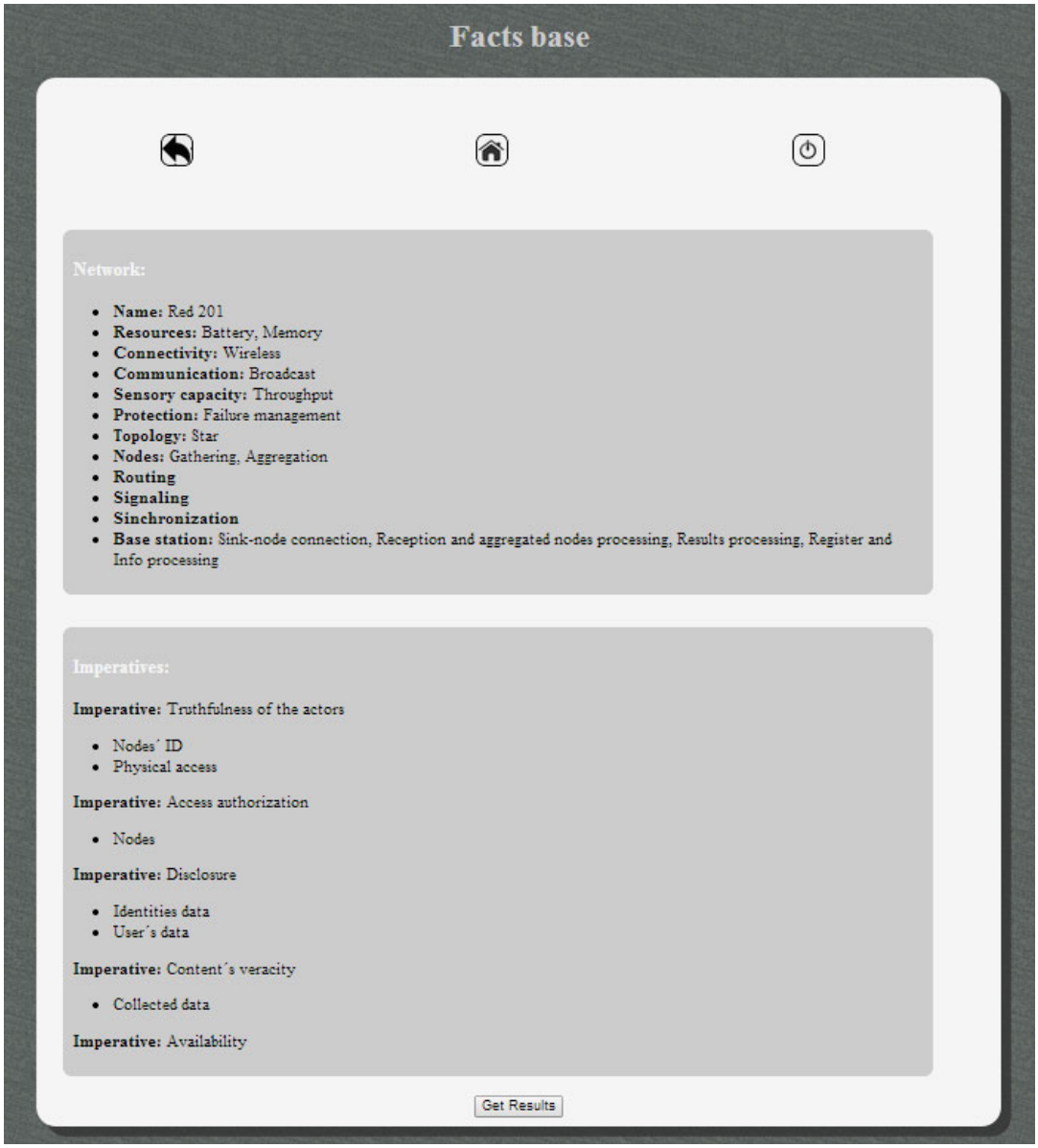


Figura 32. Vista de la Base de hechos en caso práctico

Los servicios de seguridad, tanto generales como específicos, así como los mecanismos que se necesitan implementar en este caso son los que se pueden ver en las Figuras 33 y 34, que son fragmentos de la vista de resultados12.

[12] Ver anexo "Manual de Usuario"

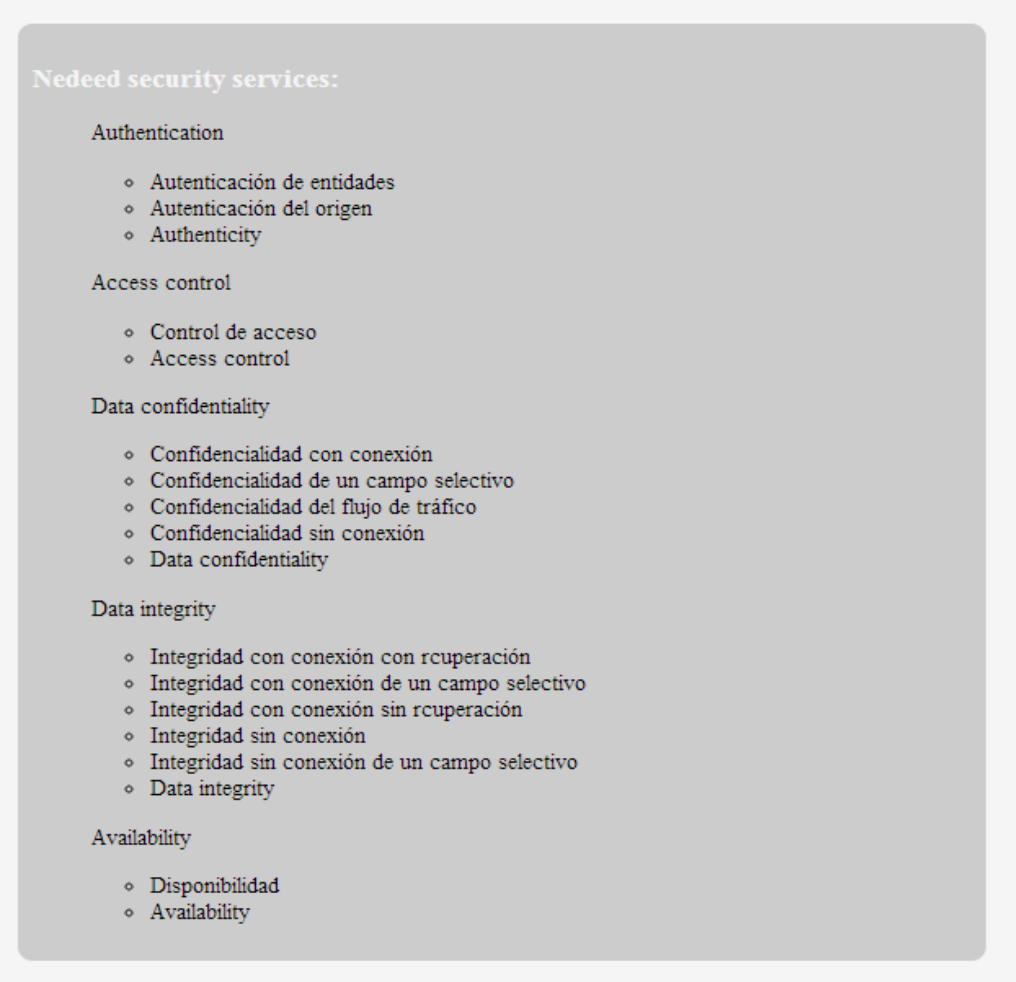


Figura 33. Servicios de seguridad (resultados caso práctico)

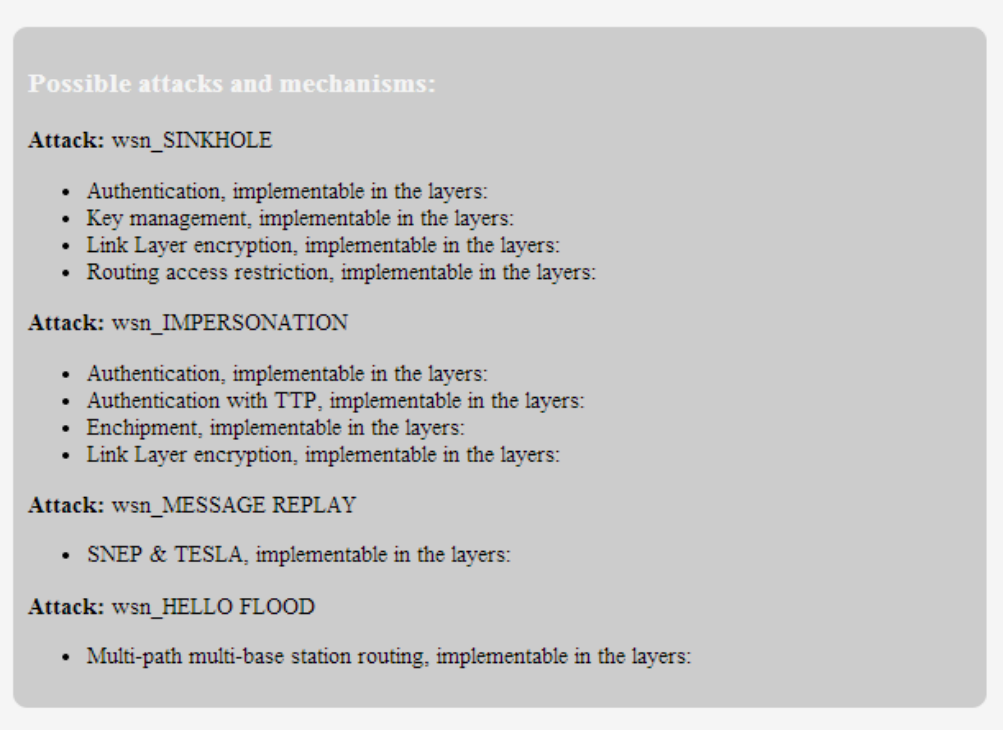


Figura 34. Ataques y mecanismos (resultados caso práctico)

CONCLUSIONES

Cómo ya se ha mencionado en la introducción, el presente proyecto perseguía el objetivo de desarrollar una aplicación que permitiera automatizar las tareas de consultorías de seguridad, que hasta la actualidad eran realizadas de forma manual, con las dificultades que esto conllevaba.

Ha sido necesario un periodo largo de estudio y también de aprendizaje de nuevas materias para la realización final de una plataforma útil y sencilla, a la que se ha denominado TSES que entiendo, tendrá aplicación en diferentes ámbitos como los de investigación, empresa, así como el cotidiano, etc.

En la fase inicial de recopilación de información ya aparecieron las primeras dificultades puesto que tanto el "Internet de las Cosas" como las Redes Inalámbricas de Sensores son tecnologías punteras que se llevan desarrollando pocos años y que por tanto, aunque es verdad que existen diversas investigaciones sobre la seguridad en estas redes, los estudios que se realizan sobre ella no están ni mucho menos centralizados e incluso en algunos artículos de revistas aparecen contradicciones.

También hubo que profundizar en el conocimiento sobre las medidas de seguridad que se pueden y se deben implantar en las Redes Inalámbricas de Sensores e intentar agrupar los dispersos conocimientos que se tienen actualmente sobre la materia, no hay que olvidar que para que el software desarrollado sea eficaz, se han de esclarecer los conceptos hasta el punto de que no exista ambigüedad posible para la máquina que los procesa, lo que conlleva gran dificultad.

Sin embargo, todo lo mencionado anteriormente ha servido para aprender nuevos conceptos que han llevado a la realización satisfactoria del objetivo principal del proyecto.

Además de la formación adquirida en esta Escuela, ha sido fundamental los conocimientos aportados en algunas de las asignaturas cursadas en la Universidad de Ciencias Aplicadas de Metropolia (Helsinki, Finlandia) y al posterior auto aprendizaje, para poder utilizar con éxito diferentes lenguajes de programación que desconocía hasta la fecha tales como: JSF, JPA, XHTML, JavaScript y CSS3 que son los que se utilizan en la actualidad para el desarrollo de aplicaciones web. Aparte, esto ha llevado a la comprensión del enorme potencial que tienen estos lenguajes y a la facilidad que se ofrece al desarrollador al utilizar dichas tecnologías modernas para así no tener la necesidad de programar a un nivel más bajo.

Al tratarse de un software de considerables dimensiones, ha resultado imprescindible profundizar en el conocimiento de las técnicas de desarrollo de software así como la realización de una planificación estructural del problema para poder enfrentarse a él de forma más efectiva. También la necesidad de recurrir a una aplicación web ha implicado la obligación de un estudio exhaustivo de la arquitectura cliente servidor.

En lo relacionado con la parte técnica, la dificultad radica en la necesidad de utilizar dos servidores y conectarlos entre sí para poder usar la aplicación en un entorno externo al del desarrollo. Curiosamente, también se han detectado algunos fallos propios de la tecnología utilizada (problemas con el CSS en las vistas de log-in y error, con la desconexión en el caso de uso de la vista log-in proporcionada por el navegador, etc) que han provocado el tener que pensar buscar otras soluciones alternativas.

Para finalizar, mencionar que el grado de conocimiento alcanzado en las WSN necesario para el desarrollo de este proyecto, se puede concluir que en los próximos años, esta tecnología se implantará de forma masiva en todos los ámbitos y sin embargo, el estudio de los mecanismos y servicios de seguridad que puedan contrarrestar los ataques que pueden sufrir estas redes es aún muy pobre por lo que se necesita una investigación más profunda de forma inminente.

TRABAJOS FUTUROS

Como cualquier software, se podrían realizar algunas mejoras que ayuden tanto que la experiencia de los usuarios sea más satisfactoria como que se ofrezcan a la salida unos resultados más precisos. Entre ellas se puede destacar:

- Ampliar la aplicación de forma que el administrador tenga más funcionalidades de extensión o reducción de la base de datos como añadir campos a las tablas, modificarlos o eliminarlos.
- Añadir la funcionalidad de que el usuario pueda insertar enormes cantidades de datos en la base de conocimientos a través de fichero y no solo manualmente a través de formularios.
- Utilizar ontologías de seguridad en vez de bases de datos relacionales para un mejor procesado de la información.

REFERENCIAS

[1] E. Fleisch, What is the Internet of Things?, Suiza: Auto-ID Lab St. Gallen, 2010.

[2] F. Mattern y C. Floerkmeier, «From the Internet of Computers to the Internet of Things,» de *From Active Data Management to Event-Based Systems and More*, Berlín, Springer Berlin Heidelberg, 2010, pp. 242-259.

[3] C. Schoenberg, «The internet of things,» *Forbes,* 2002.

[4] N. Gershenfeld, When Things Start to Think, Henry Holt and Co., 1999.

[5] E. Fleisch y F. Mattern, Das Internet der Dinge, Springer, 2005.

[6] International Telecommunication Union, «The Internet of Things,» Genova, 2005.

[7] C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern y S. Sarma, «First International Conference,» de *The Internet of Things*, 2008.

[8] DG Information Society and Media, «Networks and Communication Technologies Directorate,» de *From RFID to the Internet of Things*, Bruselas, 2006.

[9] «European Expert Conference,» de *RFID:Towards the Internet of Things*, Berlín, 2007.

[10] Commission of the European Communities, «Internet of things- An action plan for Europe,» Bruselas, 2009.

[11] S. Lange, T. Kramp y R. Kranenburg, «Introduction to the Internet of Things,» de *Enabling Things to Talk*, Berlín, Springer Berlin Heidelberg, 2013.

[12] M. Ilyas, The Handbook of Ad Hoc Wireless Networks, Florida, 2010.

[13] «Proyecto Ford,» [En línea]. Available: http://www.cse.msu.edu/~cse498/2014-01/projects/ford. [Último acceso: 3 Junio 2014].

[14] Y.-J. Wen, J. Granderson y A. M.Agogino, «Towards Embedded Wireless-Networked Intelligent Daylighting Systems for Commercial Buildings,» *IEEE,* 2006.

[15] A. Mainwaring., J. Polastre., R. Szewcyk., D. Culler. y J. Anderson., «Wireless Sensor Networks for Habitat Monitoring,» de *Local Computer Networks*, Denver, 2010.

[16] K. Martinez, P. Padhy, A. Riddoch, H. Ong y J. Hart, «Glacial Environment Monitoring using Sensor Networks,» de *Real-World Wireless Sensor Networks*, Estocolmo, 2005.

[17] L. Evers, M. Bijl, M. Marin-Perianu, R. Marin-Perianu y P. Havinga, «Wireless Sensor Networks and Beyond: A Case Study on Transport and Logisitics,» 2005.

[18] M. Morris y J. Lundell, «Ubiquitous Computing for Cognitive Decline,» 2003.

[19] «In Dust We Trust,» *The Economist,* vol. 371, nº 8379, pp. 10-12.

[20] U. S. Customs and Border Protection, «SBInet Program,» 2009.

[21] «Camalie Net Wireless Sensing,» [En línea]. Available: http://camalie.com. [Último acceso: 3 Junio 2014].

[22] A. Prati, R. Vezzani, L. Benini, E. Farella y P. Zappi, «An An integrated multi-modal sensor network for video surveillance,» 2005.

[23] M. Thomas, E. Peytchev y D. Al-Dabass, «Auto-sensing and distribution of traffic information in vehicular ad hoc networks,» *International Journal of Simulation Systems, Science & Technology,* vol. 5, pp. 59-63, 2004.

[24] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser y M. Turon., «Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks,» *Information Processing in Sensor Networks,* pp. 254-263, 2007.

[25] D. Baley y E. Wright, Practical SCADA FOR INDUSTRY, Gran Bretaña: Elsevier, 2003.

[26] Office of the Manager Communication System, «Supervisory Control and Data Acquisition (SCADA) Systems,» 2004.

[27] S. Gupta, H. K. Verma y A. L. Sangal, «Security Attacks & Prerequisite for Wireless Sensor Networks,» *International Journal of Engineeering and Advanced Technology (IJEAT),* vol. 2, pp. 558-566, 2013.

[28] E. Yvette, S. Jeon y T. Kim, «IPv6 Mobile Sensor Network Architecture for SCADA System,» de *International Conference on Computational Intelligence and Communication Systems*, 2011.

[29] R. McClanahan, «The Benefits of Networked SCADA Systems Utilizing IP-Enabled Networks,» de *Rural Electric Power Conference*, 2002.

[30] J. A. Sánchez, L. López, J.-F. Martínez y P. Castillejo, «Automated determination of security services to ensure personal data protection in the Internet of Things applications,» de *Third International Conference on Innovative Computing Technology*, Londres, 2013.

[31] L. E. Palafox y J. A. Garcia-Macias, Security in Wireless Sensor Networks, Méjico, 2008.

[32] H. Alzaid, D. Park, J. G. Nieto, C. Boyd y E. Foo, «A Forward and Backward Secure Key Management in Wireless Sensor Networks for PCS/SCADA,» de *First International ICST Conference*, 2009.

[33] Y. Wang, G. Attebury y B. Ramamurthy, «A Survey of Security Issues in Wireless Sensor Networks,» *IEEE Commun. Surveys & Tutorials,* vol. 8, pp. 2-23, 2006.

[34] P. J. Walters, Z. Liang, W. Shi y V. Chaudhary, «Wireless Sensor Network Security: A Survey,» *IEEE Commun. Surveys & Tutorials,* pp. 52-73, 2009.

[35] A. D. Wood y J. A. Stankovic, «Denial of service in sensor networks,» *Computer,* pp. 54-62, 2002.

[36] J. Deng, R. Han y S. Mishra, «Intrusion-tolerant routing in wireless sensor networks,» Department of Computer Science, Universidad de Colorado, Colorado, 2002.

[37] C. Ozturk, Y. Zhang y W. Trappe, «Source-Location Privacy in Energy-Constrained Sensor Network Routing,» 2004.

[38] X. Wang, W. Gu, S. Chellappan, D. Xuan y T. H. Laii, «Search-based physical attacks in sensor networks: Modeling and defense,» Ohio, 2005.

[39] ITU, *Recomendación X.800: Arquitectura de seguridad de la interconexión de sistemas abiertos para aplicaciones del CCITT,* 2003.

[40] CITSEM, «Proyecto AWARE,» [En línea]. Available: https://www.citsem.upm.es/index.php/proyectos-es?view=project&task=show&id=48.

[41] «Java Enterprise,» [En línea]. Available: http://docs.oracle.com/javaee/6/tutorial/doc/bnaaw.html. [Último acceso: 3 Junio 2014].

[42] «Especificación de Java Server Faces,» [En línea]. Available: https://www.jcp.org/en/jsr/detail?id=3. [Último acceso: 3 Junio 2014].

[43] «Especificación de Java Persistence,» [En línea]. Available: https://www.jcp.org/en/jsr/detail?id=220. [Último acceso: 3 Junio 2014].

[44] «Glassfish,» [En línea]. Available: https:/glassfish.java.net/es/. [Último acceso: 3 Junio 2014].

[45] *Ley Orgánica de Protección de Datos de Carácter Personal,* 1999.

ANEXOS

ANEXO I - MANUAL DE USUARIO

Seguridad

En el caso del GlassFish, la consola de configuración se encuentra en el puerto 4848 y una vez dentro, se ha de ir a la pestaña "configuración" y se ha de llegar a la pestaña "file" como se muestra en la Figura 35.

Figura 35. Pestaña file

En esa pantalla, se hace click en el botón "Gestionar Usuarios" situado en la parte superior derecha como se muestra en la Figura 36.

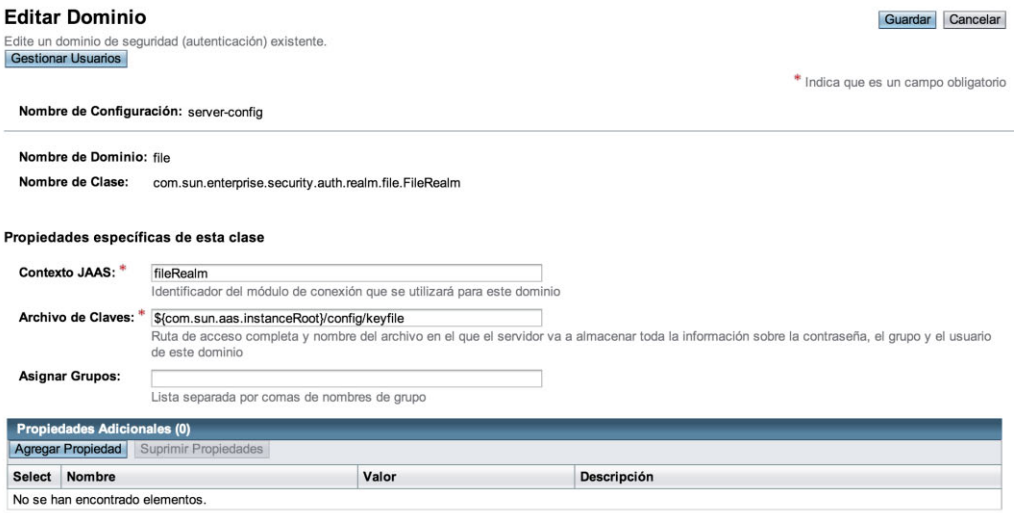


Figura 36. Figura File

Finalmente aparecerá una vista donde se pueden realizar todas las acciones indicadas anteriormente.

Figura 37. Gestionar usuarios

Es importante que al finalizar con la gestión de usuarios en la pantalla de la Figura 37 se guarde la información y para ello se ha de hacer click primero en el botón "Atrás" de la Figura 37 y posteriormente al botón "Guardar" de la Figura 36.

Por último, en la pestaña "Seguridad" que aparece en la Figura 35, se debe de seleccionar la opción "Asignación de Principal por Defecto a Rol" como se observa en la Figura 38.

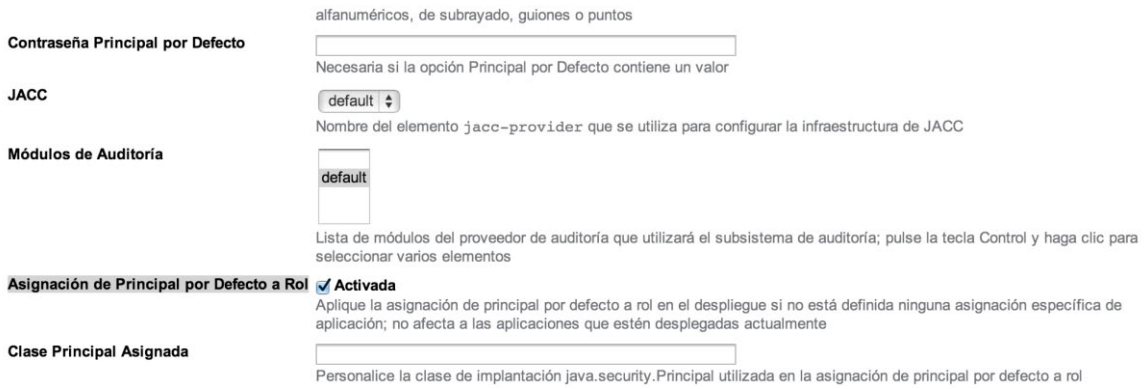


Figura 38. Asignación por defecto a rol

Vistas

La aplicación web se compone de un total de treinta y una páginas web. La mayoría de ellas comparten elementos en común, como son los botones de navegación (atrás, menú principal y cerrar sesión) que se sitúan en la parte superior de la vista o también la fecha y hora en la que el usuario ha accedido a dicha vista que se sitúa en la parte inferior por ejemplo. Pero todas y cada una de ellas tienen una función específica dentro del entramado de esta aplicación web.

En la Figura 39 se puede observar la situación de los botones de navegación que comparten todas las vistas excepto las de error, la vista de punto de partida y el menú principal.

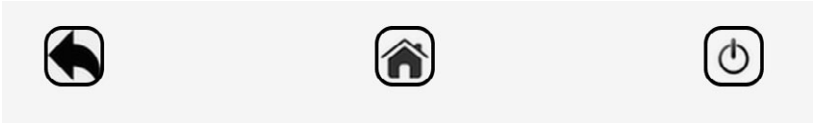


Figura 39. Botones de navegación

De izquierda a derecha en la Figura 39 se sitúan el botón de retroceso que al hacer click en él volverá a la vista anterior, el botón de home que devolverá directamente la vista del menú principal y el botón de desconexión que cerrará la sesión abierta en ese momento y volverá a la vista del punto de partida.

1 Punto de partida

La Figura 40, es la correspondiente al "índex" de cualquier aplicación web. No tiene una función en particular, sino, simplemente es la portada de la aplicación en la que el usuario puede empezar a interactuar con la misma. También se incluye el escudo de la ETSIST, con la ventaja de que al hacer click en él, se abrirá una nueva pestaña en el navegador con la página web de la escuela.

Generalmente es la predecesora al proceso de "log in" ya que cuando el usuario pulse el botón "Start" se redirigirá directamente a la Figura 40 en el caso de que no se haya autenticado con anterioridad.

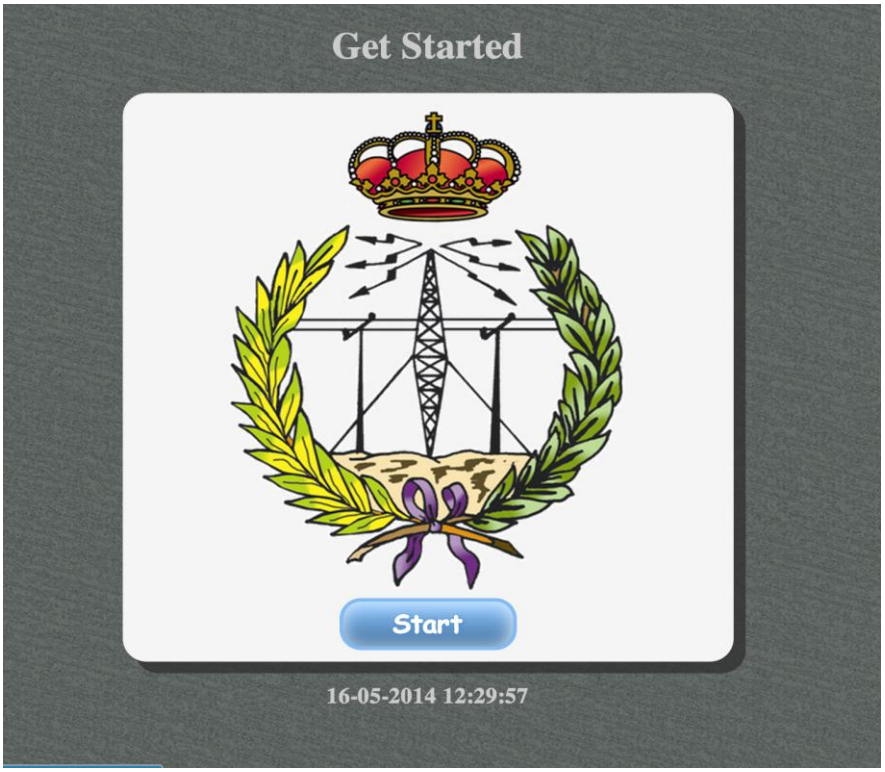


Figura 40. Punto de partida

2 Log in

La vista 3 aparecerá normalmente cuando el usuario haga click en el botón “Start” de la vista del punto de partida pero también puede ocurrir que se produzca una desconexión de sesión pasiva por parte del servidor por la finalización del tiempo máximo de inactividad de la sesión.

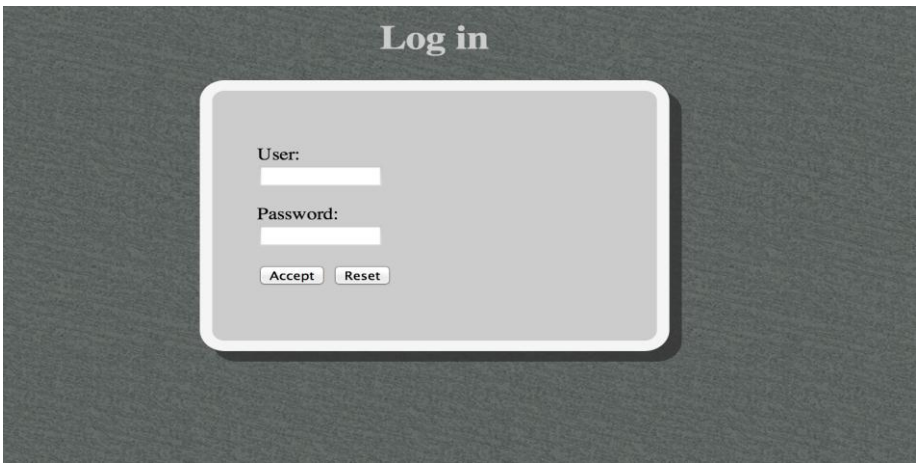


Figura 41. Log in

3 Menú principal

Como su propio nombre indica, es el "eje" de la aplicación ya que a partir del mismo se puede navegar en diferentes direcciones dependiendo de cuál sea el objetivo de la sesión.

Se puede observar en la Figura 42, que las opciones que ofrece este menú al usuario son: evaluación por fichero, evaluación rápida, inspeccionar base de conocimientos, modificar base de conocimientos, revisar/descargar el log y por último el apartado de ¿Quiénes somos?.

En el caso del menú principal, los botones de navegación que se pueden ver son el de menú principal que lo único que haría es recargar la página de nuevo ya que redirige a la misma vista y el botón de cerrar sesión que una vez realizada la acción redirigiría al usuario al punto de inicio de nuevo.

Figura 42. Menú principal.

4 Evaluación por fichero

Está pensada para que cualquier usuario que quiera proceder a realizar una evaluación, la pueda hacer directamente subiendo un fichero que, con un formato determinado por una serie de normas, permita a la aplicación interpretar los datos igual que si fueran introducidos manualmente desde la pantalla.

En este caso, a diferencia de la vista “menú principal”, aparece un botón de “atrás” que en esta ocasión haría lo mismo que el botón de “home” o “menú principal”.

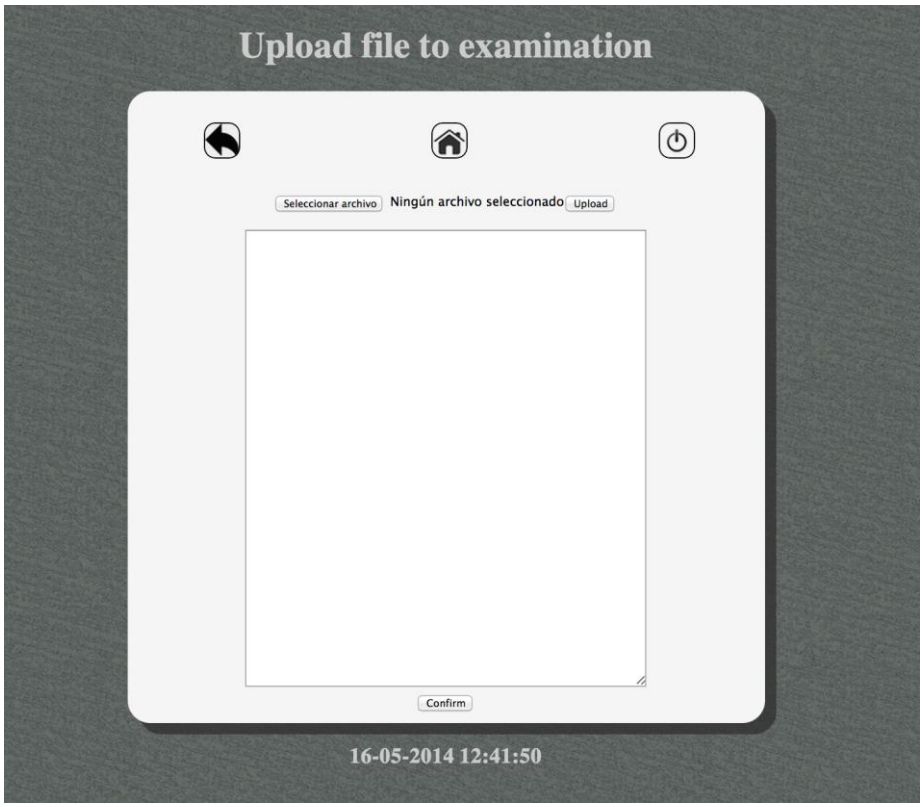


Figura 43. Evaluación por fichero parte 1

Lo primero que se haría en esta vista es dar al botón “seleccionar archivo”. Una vez seleccionado el fichero “txt” que queramos subir al servidor, se debe de hacer click en “Upload” apareciendo la Figura 44.

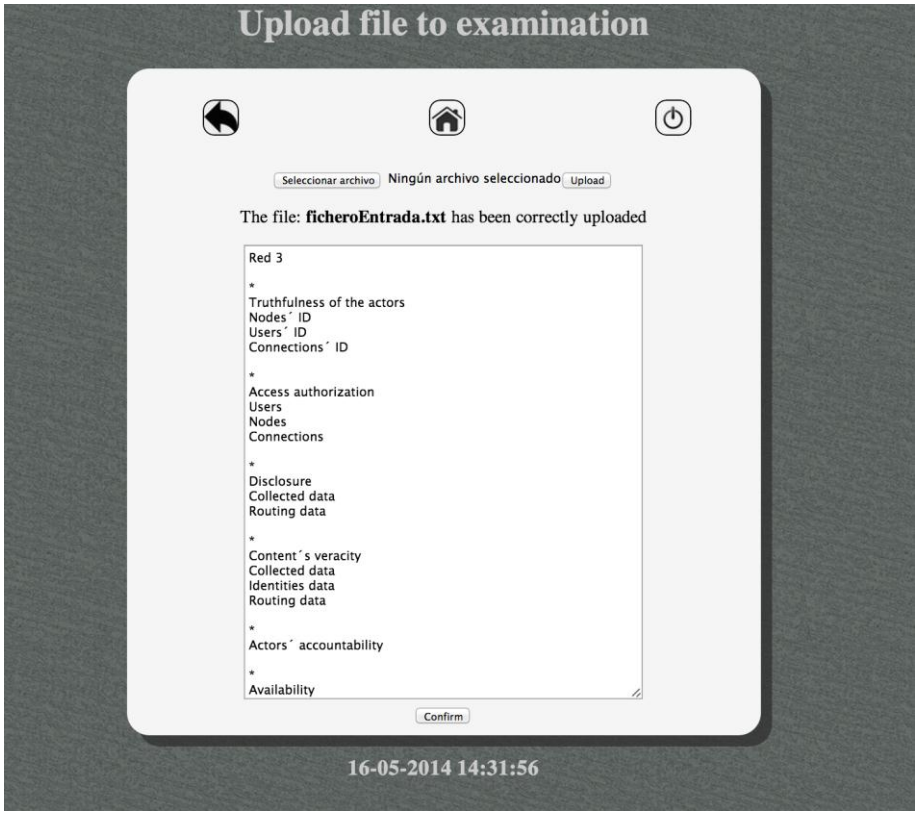


Figura 44. Evaluación por fichero parte 2

Como se explicará en otro apartado, éste es el resultado del caso en que no haya habido errores en la lectura. En cualquier otro caso aparecerían sus correspondientes mensajes de error en el lugar donde aquí aparece la frase "The file: ficheroEntrada.txt has been correctly uploaded". También se reflejaría toda la información, tanto de error como de éxito en el log.

Una vez subido correctamente se procede pinchando el botón de confirmar y en el caso de que el fichero esté bien formado y de que se haya podido leer correctamente la información pasará a la vista de "Base de hechos" que se verá posteriormente. En caso de que el fichero tuviera errores tipográficos, se notificará el tipo de error y la línea tanto por pantalla como en el log.

5 Evaluación Rápida

Esta es la otra posibilidad de evaluación donde el usuario va escogiendo dinámicamente qué imperativos y datos a los que se aplican dichos imperativos quiere introducir en la lógica del programa. Para ello se ofrece la Figura 45.

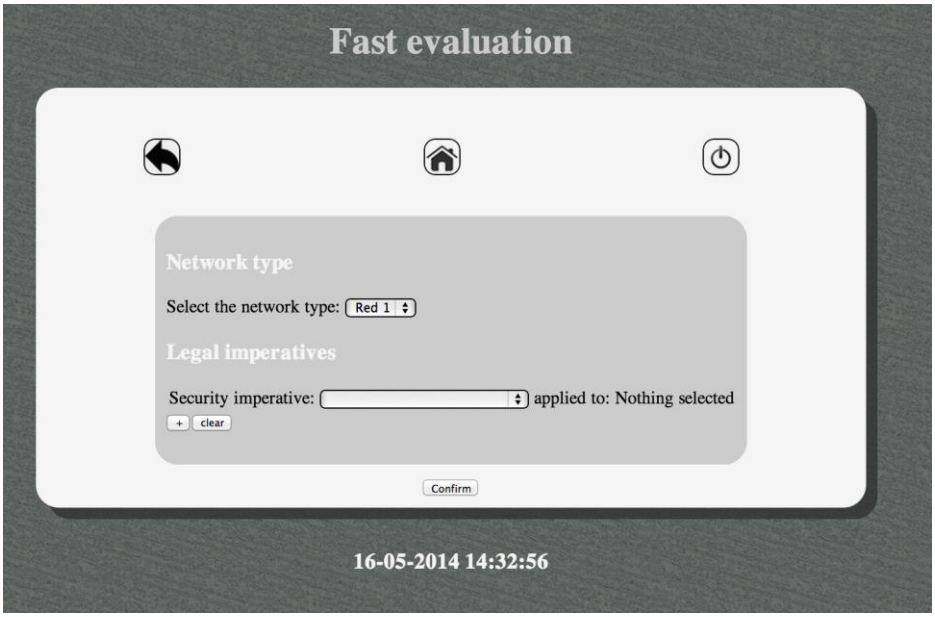


Figura 45. Evaluación rápida parte 1

Primero se ha de elegir el tipo de red con el que se quiere tratar. La aplicación dará única y exclusivamente la posibilidad de elegir entre los tipos de redes que se encuentren en la base de conocimientos.

Una vez elegido el tipo de red se ha de proceder eligiendo el imperativo de seguridad. Con ello, dependiendo del imperativo escogido, aparecerán las opciones de todos los elementos a los que se puede aplicar ese imperativo de los cuales se deberá escoger uno.

Para elegir el siguiente imperativo, se pulsa el botón "+" y si se quiere volver a empezar desde el principio se pulsaría "clear".

En la Figura 46 aparece un ejemplo de cómo quedaría listo para el siguiente paso:

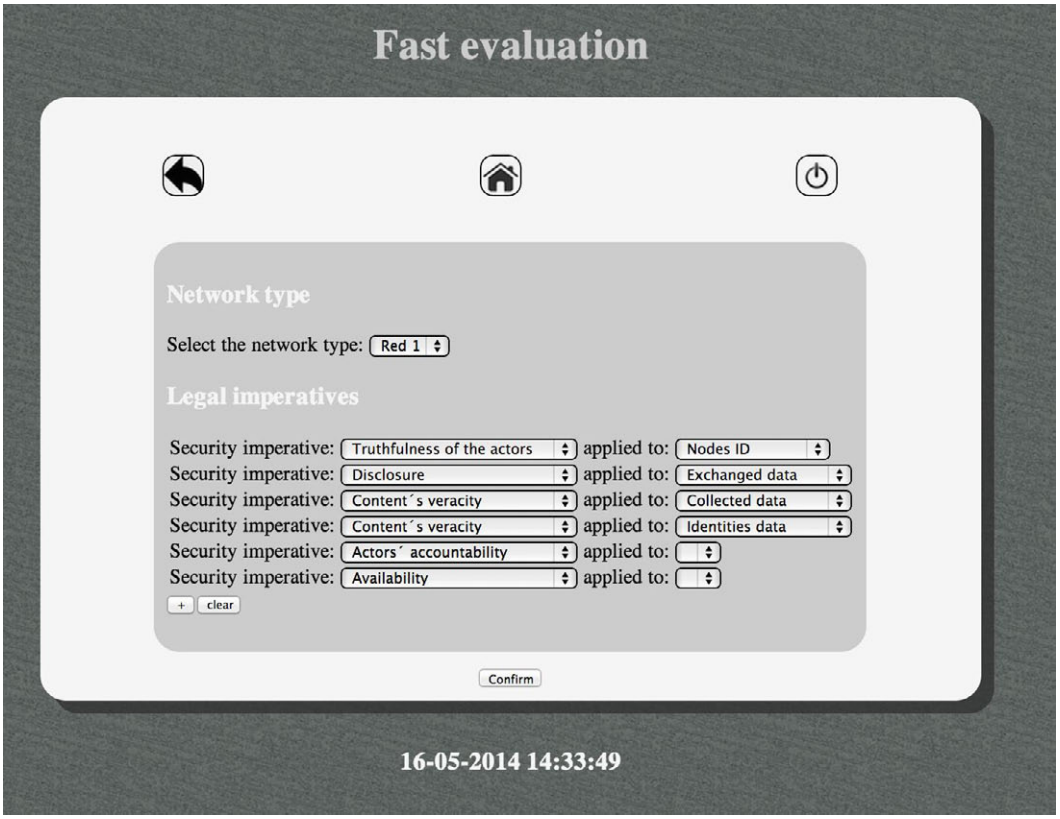


Figura 46. Evaluación rápida parte 2

A partir de aquí se hace click en “Confirm” y se pasa a la vista “Base de hechos” que se explicará posteriormente.

6 Inspeccionar base de conocimientos

Este apartado está pensado para el usuario que está interesado en examinar algún concepto en concreto relacionado con la base de datos o buscar en la misma algún ataque, mecanismo, servicio o red para chequear sus características.

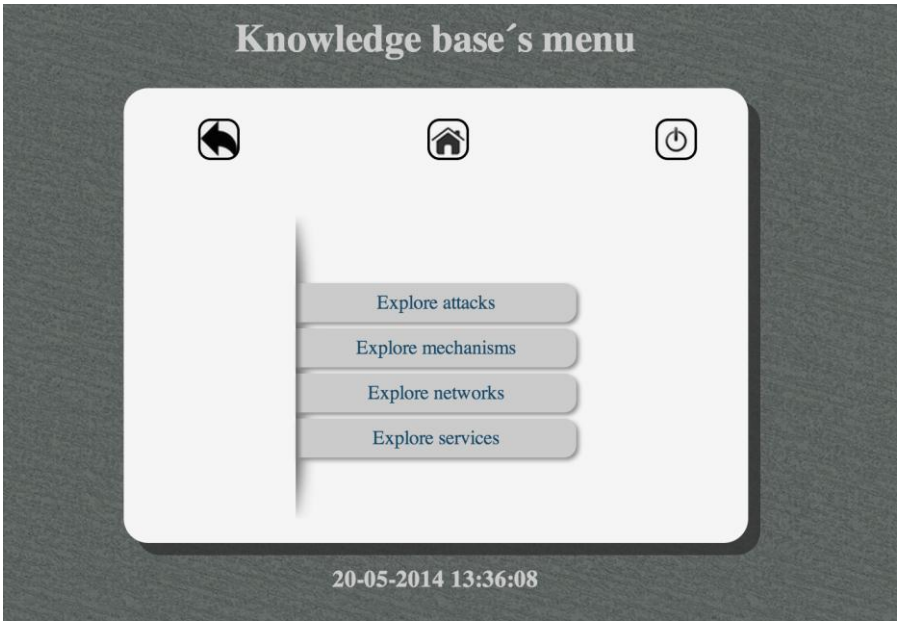


Figura 47. Menú Inspeccionar base de conocimientos

Como se puede observar en la Figura 47, este menú ofrece cuatro posibilidades que corresponden con las principales tablas de conocimientos en las que basa su lógica de programa la aplicación web.

6.1 Inspeccionar Ataques

Como se ha dicho con anterioridad, el usuario puede: buscar uno o varios ataques por alguna de sus características y también ordenar los ataques de la tabla por una característica.

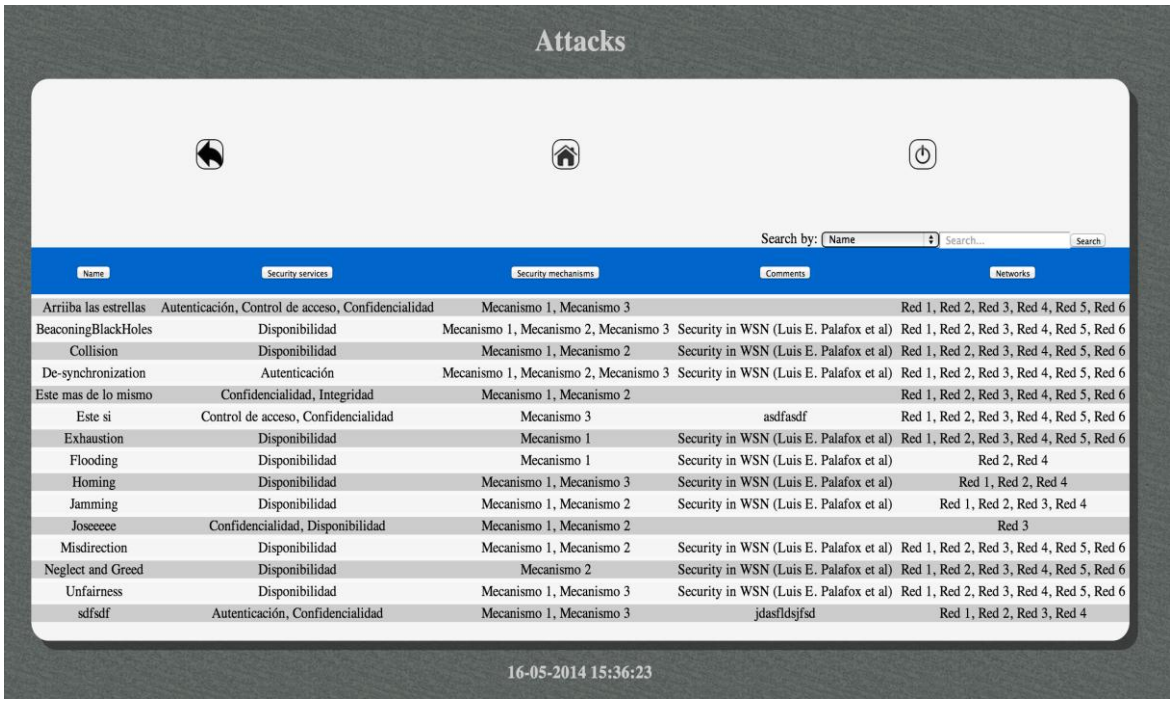


Figura 48. Inspeccionar ataques

6.2 Inspeccionar Mecanismos

El usuario puede: buscar uno o varios mecanismos por alguna de sus características y también ordenar los mecanismos de la tabla por una característica.

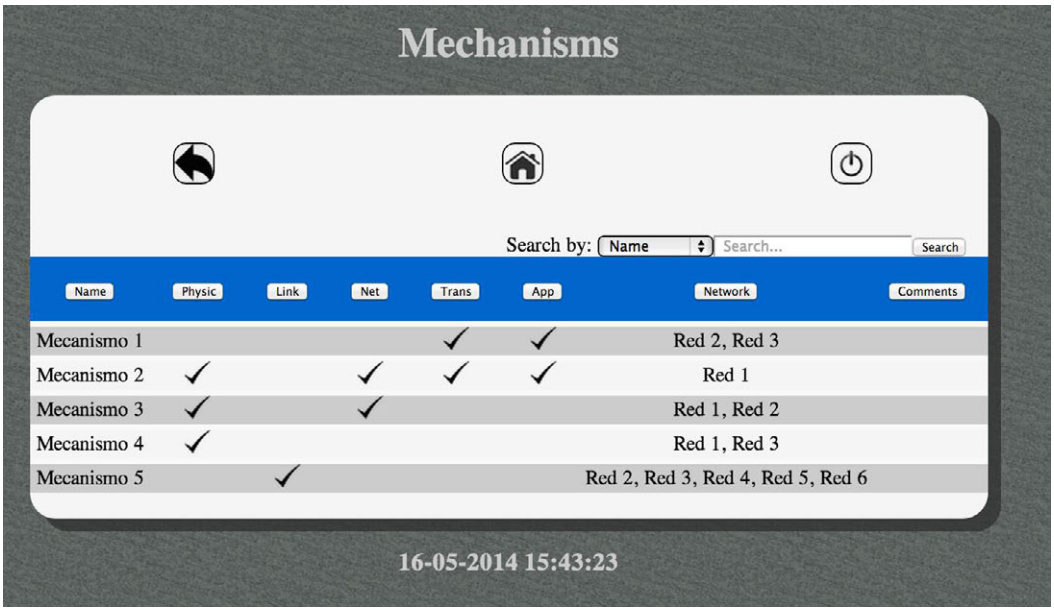
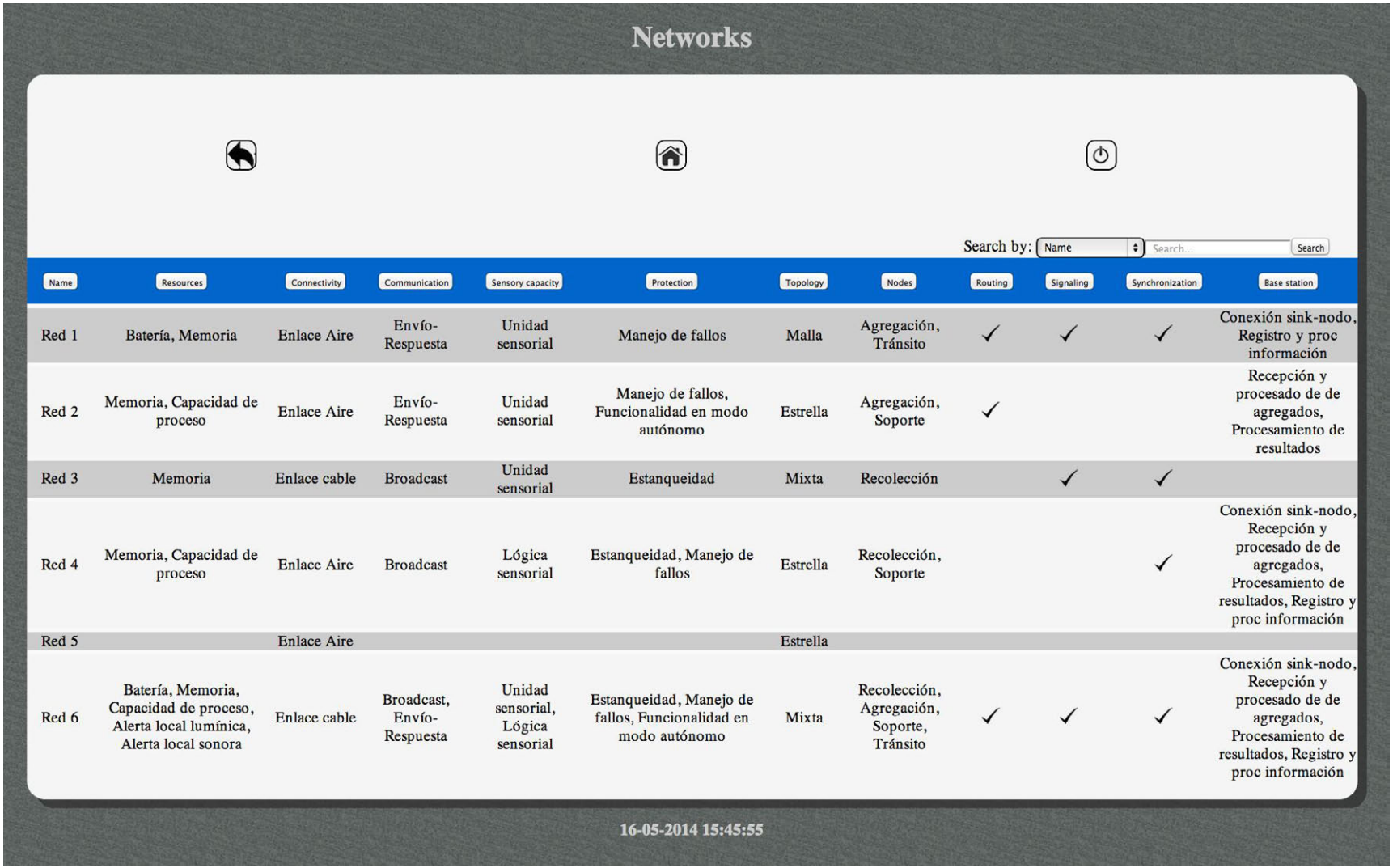


Figura 49. Inspeccionar mecanismos

6.3 Inspeccionar Redes

El usuario puede: buscar una o varias redes por alguna de sus características y también ordenar las redes de la tabla por una característica.



Name	Resources	Connectivity	Communication	Sensory capacity	Protection	Topology	Nodes	Routing	Signaling	Synchronization	Base station
Red 1	Batería, Memoria	Enlace Aire	Envío-Respuesta	Unidad sensorial	Manejo de fallos	Malla	Agregación, Tránsito	✓	✓	✓	Conexión sink-nodo, Registro y proc información
Red 2	Memoria, Capacidad de proceso	Enlace Aire	Envío-Respuesta	Unidad sensorial	Manejo de fallos, Funcionalidad en modo autónomo	Estrella	Agregación, Soporte	✓			Recepción y procesado de de agregados, Procesamiento de resultados
Red 3	Memoria	Enlace cable	Broadcast	Unidad sensorial	Estanqueidad	Mixta	Recolección		✓	✓	
Red 4	Memoria, Capacidad de proceso	Enlace Aire	Broadcast	Lógica sensorial	Estanqueidad, Manejo de fallos	Estrella	Recolección, Soporte			✓	Conexión sink-nodo, Recepción y procesado de de agregados, Procesamiento de resultados, Registro y proc información
Red 5		Enlace Aire				Estrella					
Red 6	Batería, Memoria, Capacidad de proceso, Alerta local lumínica, Alerta local sonora	Enlace cable	Broadcast, Envío-Respuesta	Unidad sensorial, Lógica sensorial	Estanqueidad, Manejo de fallos, Funcionalidad en modo autónomo	Mixta	Recolección, Agregación, Soporte, Tránsito	✓	✓	✓	Conexión sink-nodo, Recepción y procesado de de agregados, Procesamiento de resultados, Registro y proc información

Figura 50. Inspeccionar redes

6.4 Inspeccionar Servicios

Es la última de este grupo, el usuario puede: buscar y ordenar la tabla por servicio general o servicio específico.

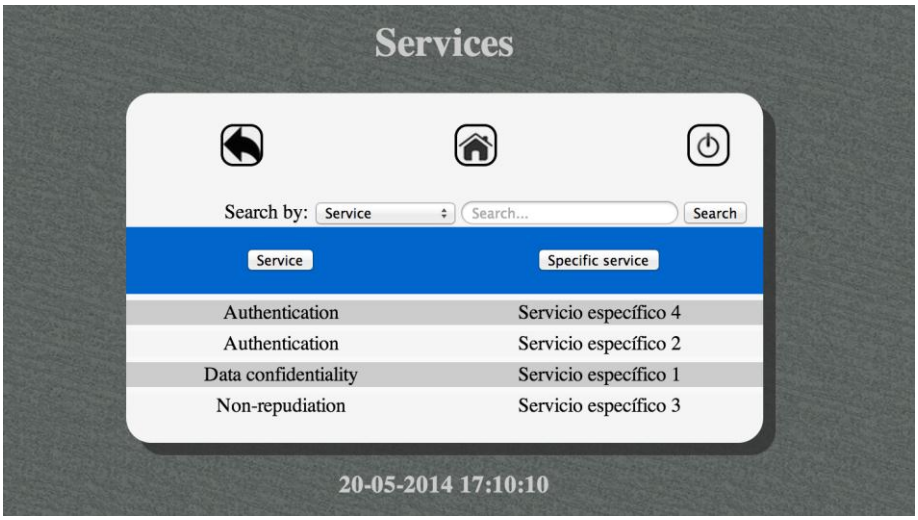


Figura 51. Inspeccionar servicios

7 Modificar base de conocimientos

Este grupo de vistas están pensadas única y exclusivamente para los usuarios pertenecientes al grupo administrador, es decir, para aquellos que siendo expertos de seguridad tengan los conocimientos suficientes para poder modificar, añadir o borrar los datos que se presentan en la base de conocimientos y que la lógica del programa utilizará para determinar su salida.

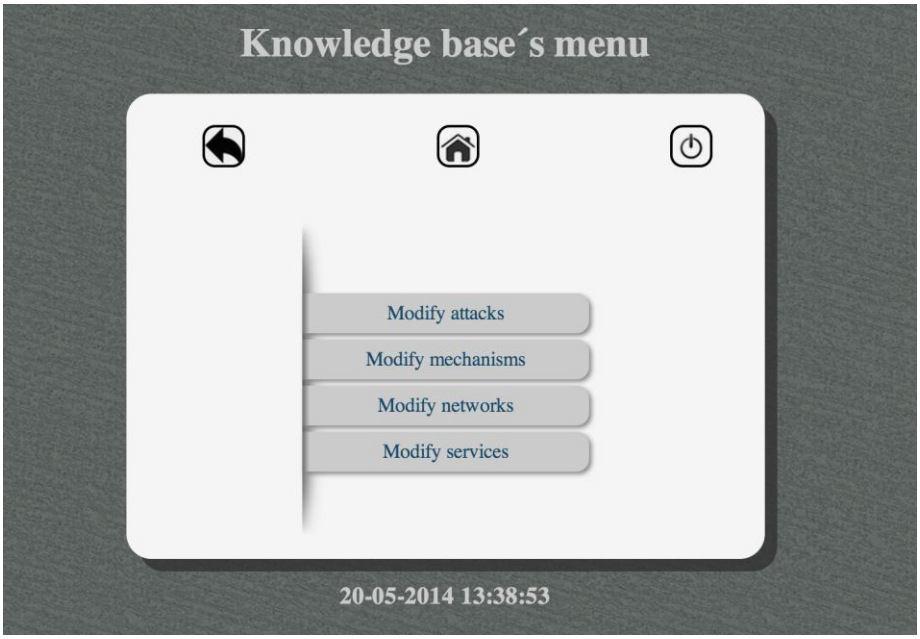


Figura 52. Modificar base de conocimiento

7.1 Modificar Ataques

Es prácticamente igual a la vista "Inspeccionar Ataques" excepto por un botón para agregar un nuevo ataque y por otros dos botones que se ofrecen al final de la tabla para cada uno de los ataques y que permitirán al administrador tanto modificar como borrar ese determinado ataque. Al igual que en la vista de inspeccionar, también se ofrecen las posibilidades de buscar un determinado ataque por una de sus características y ordenar la tabla por alguna de sus columnas.

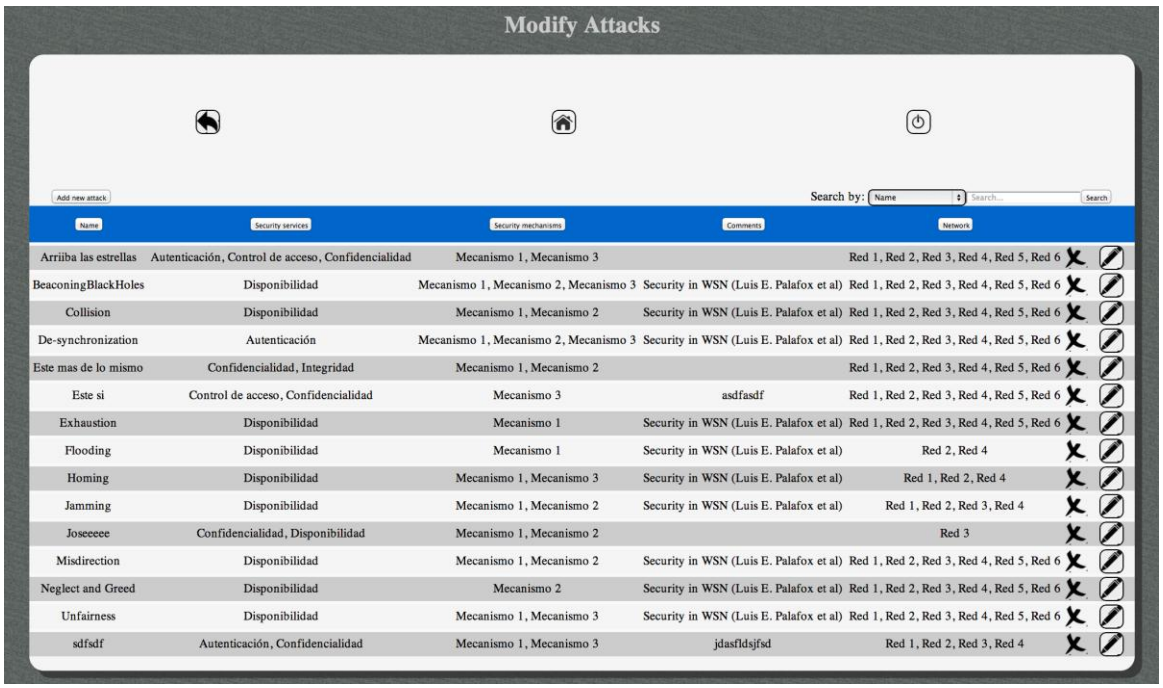


Figura 53. Modificar ataques

Cuando el usuario desea agregar un ataque nuevo a la base de conocimientos, sólo debe hacer click en el botón “Add new attack” y esto le llevará a la siguiente pantalla:

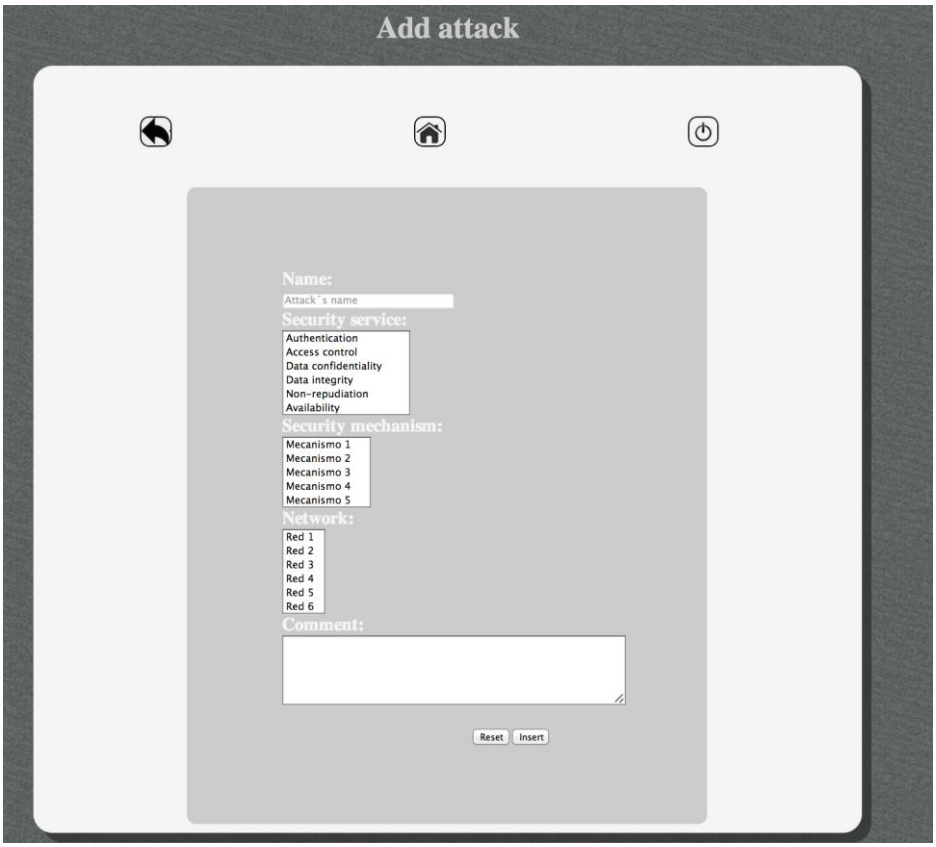


Figura 54. Añadir un nuevo ataque

Como se observa en la vista 13, para añadir un nuevo ataque, el usuario ha de seleccionar un nombre que no esté anteriormente en la base de conocimientos porque existe un mecanismo de validación que comprobará dicho aspecto y en ese caso será avisado por pantalla al usuario. Posteriormente deberá escoger los servicios de seguridad a los que ese ataque afecta, los mecanismos necesarios para paliar el ataque y las redes a las que puede afectar el ataque.

Todos los campos indicados anteriormente son obligatorios por lo que en los que se ofrecen multi-selección, se debe de escoger al menos uno. También se puede introducir optativamente un comentario de hasta 400 caracteres.

Finalmente para agregar el ataque se procederá haciendo click en el botón “Insert”.

En el caso de que se quiera modificar un ataque ya existente en la tabla de ataques de la base de conocimientos se procederá a hacer click en el botón con forma de lápiz situado en la última columna de la tabla para cada uno de los ataques. Como resultado aparecerá la siguiente vista:

Figura 55. Modificar ataque

El funcionamiento y la validación son mismos que para el caso de añadir un nuevo ataque con la única salvedad de que las características que tenía el ataque en la base de conocimientos, aparecerán ya seleccionadas.

Por último es importante apuntar que tanto las opciones de selección de los mecanismos y las redes presentados en la pantalla proceden directamente de las entradas que existen en las tablas de mecanismos y redes respectivamente en la base de conocimientos de la aplicación.

7.2 Modificar Mecanismos

Es prácticamente igual a la vista “Inspeccionar Mecanismos” excepto por un botón para agregar un nuevo mecanismo y por otros dos botones que se ofrecen al final de la tabla para cada uno de los mecanismos y que permitirán al administrador tanto modificar como borrar ese determinado mecanismo. Al igual que en la vista de inspeccionar, también se ofrecen las posibilidades de buscar un determinado mecanismo por una de sus características y ordenar la tabla por alguna de sus columnas.

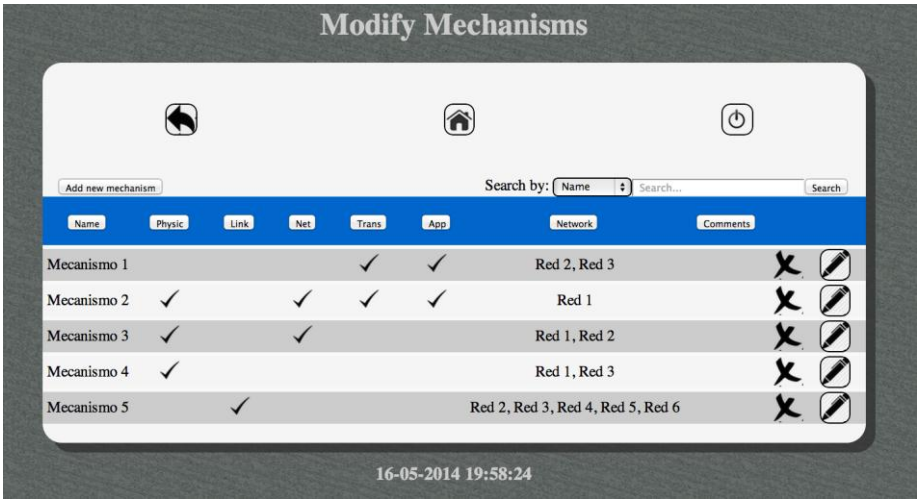


Figura 56. Modificar mecanismos

Al igual que en el caso de que se quiera añadir un nuevo ataque, para añadir un nuevo mecanismo se debe de hacer click en el botón "Add new mechanism" que llevará a la siguiente vista:

Figura 57. Añadir nuevo mecanismo

El funcionamiento prácticamente es el mismo que al añadir un ataque ya que el nombre del mecanismo no puede ser repetido en la base de datos y las redes que se ofrecen son aquellas que existan en la base de datos. El apartado de comentarios también es opcional y su longitud máxima es de 400 caracteres.

De la misma forma que en la parte de ataques, en el caso de que se quiera modificar algún mecanismo se ha de hacer click en el botón con forma de lápiz y se accederá directamente a la siguiente vista:

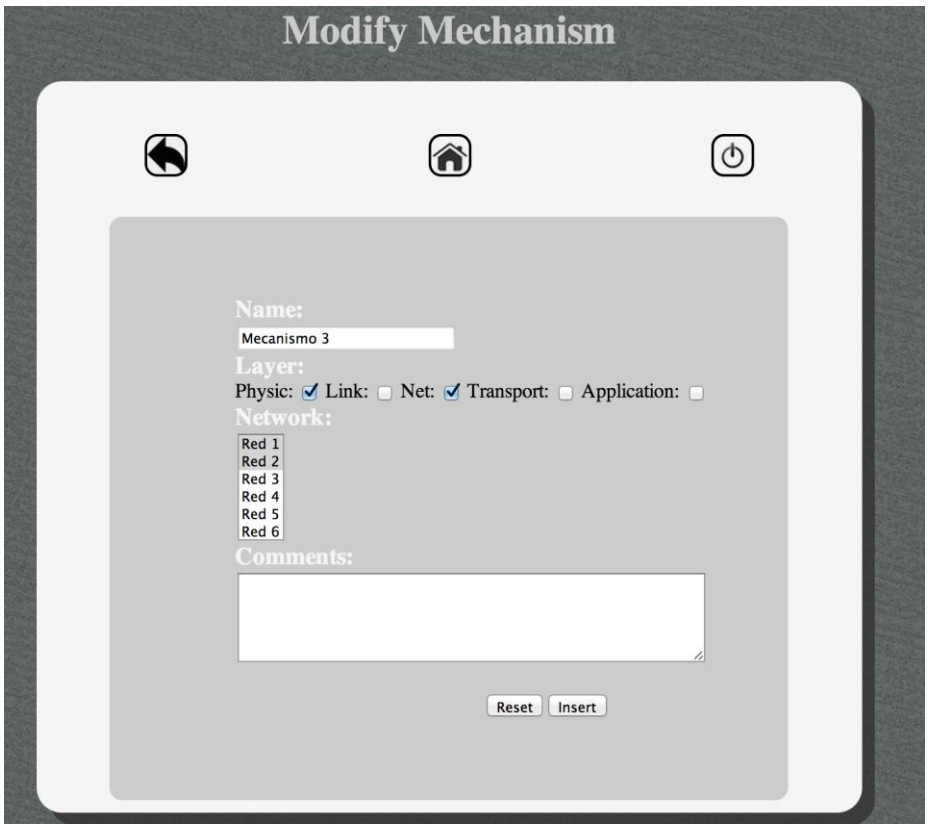
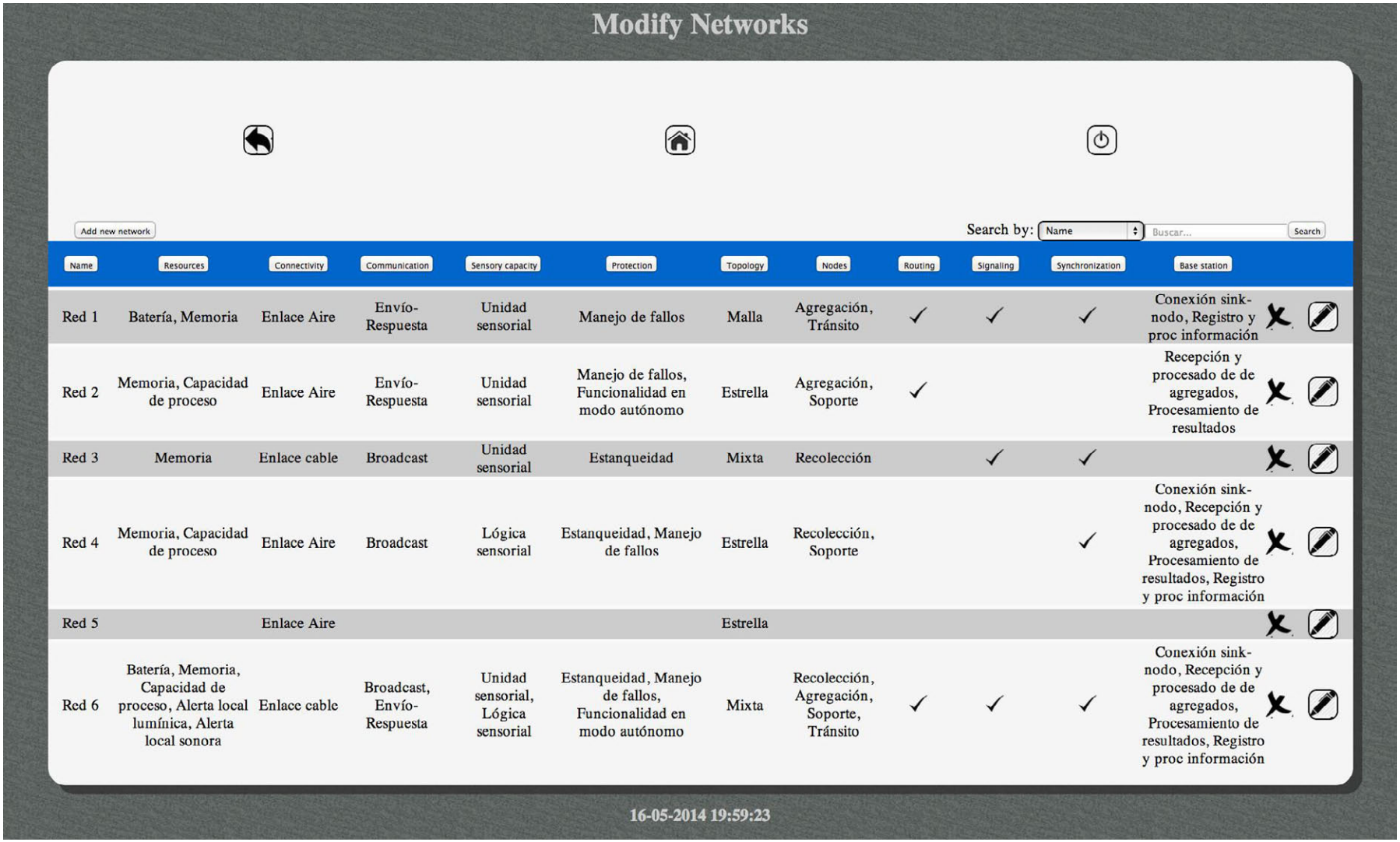


Figura 58. Modificar mecanismo

Como se observa en la vista 20, las características que tenía previamente el mecanismo aparecerán reflejadas de nuevo al intentar modificar el mecanismo. La validación y el funcionamiento es el mismo que para añadir un nuevo mecanismo.

7.3 Modificar Redes

Es prácticamente igual a la vista "Inspeccionar Redes" excepto por un botón para agregar una nueva red y por otros dos botones que se ofrecen al final de la tabla para cada una de las redes y que permitirán al administrador tanto modificar como borrar esa determinada red. Al igual que en la vista de inspeccionar, también se ofrecen las posibilidades de buscar una determinada red por una de sus características y ordenar la tabla por alguna de sus columnas.



Name	Resources	Connectivity	Communication	Sensory capacity	Protection	Topology	Nodes	Routing	Signaling	Synchronization	Base station
Red 1	Batería, Memoria	Enlace Aire	Envío-Respuesta	Unidad sensorial	Manejo de fallos	Malla	Agregación, Tránsito	✓	✓	✓	Conexión sink-nodo, Registro y proc información
Red 2	Memoria, Capacidad de proceso	Enlace Aire	Envío-Respuesta	Unidad sensorial	Manejo de fallos, Funcionalidad en modo autónomo	Estrella	Agregación, Soporte	✓			Recepción y procesado de de agregados, Procesamiento de resultados
Red 3	Memoria	Enlace cable	Broadcast	Unidad sensorial	Estanqueidad	Mixta	Recolección		✓	✓	
Red 4	Memoria, Capacidad de proceso	Enlace Aire	Broadcast	Lógica sensorial	Estanqueidad, Manejo de fallos	Estrella	Recolección, Soporte			✓	Conexión sink-nodo, Recepción y procesado de de agregados, Procesamiento de resultados, Registro y proc información
Red 5		Enlace Aire				Estrella					
Red 6	Batería, Memoria, Capacidad de proceso, Alerta local lumínica, Alerta local sonora	Enlace cable	Broadcast, Envío-Respuesta	Unidad sensorial, Lógica sensorial	Estanqueidad, Manejo de fallos, Funcionalidad en modo autónomo	Mixta	Recolección, Agregación, Soporte, Tránsito	✓	✓	✓	Conexión sink-nodo, Recepción y procesado de de agregados, Procesamiento de resultados, Registro y proc información

Figura 59. Modificar redes

Como ya se ha explicado anteriormente, para añadir una nueva red se hace click en el botón “Add new network”. Y para modificar también se procedería de la misma forma que en los dos casos anteriores.

Las vistas 22 y 23 muestran los formularios para añadir y modificar una red respectivamente.

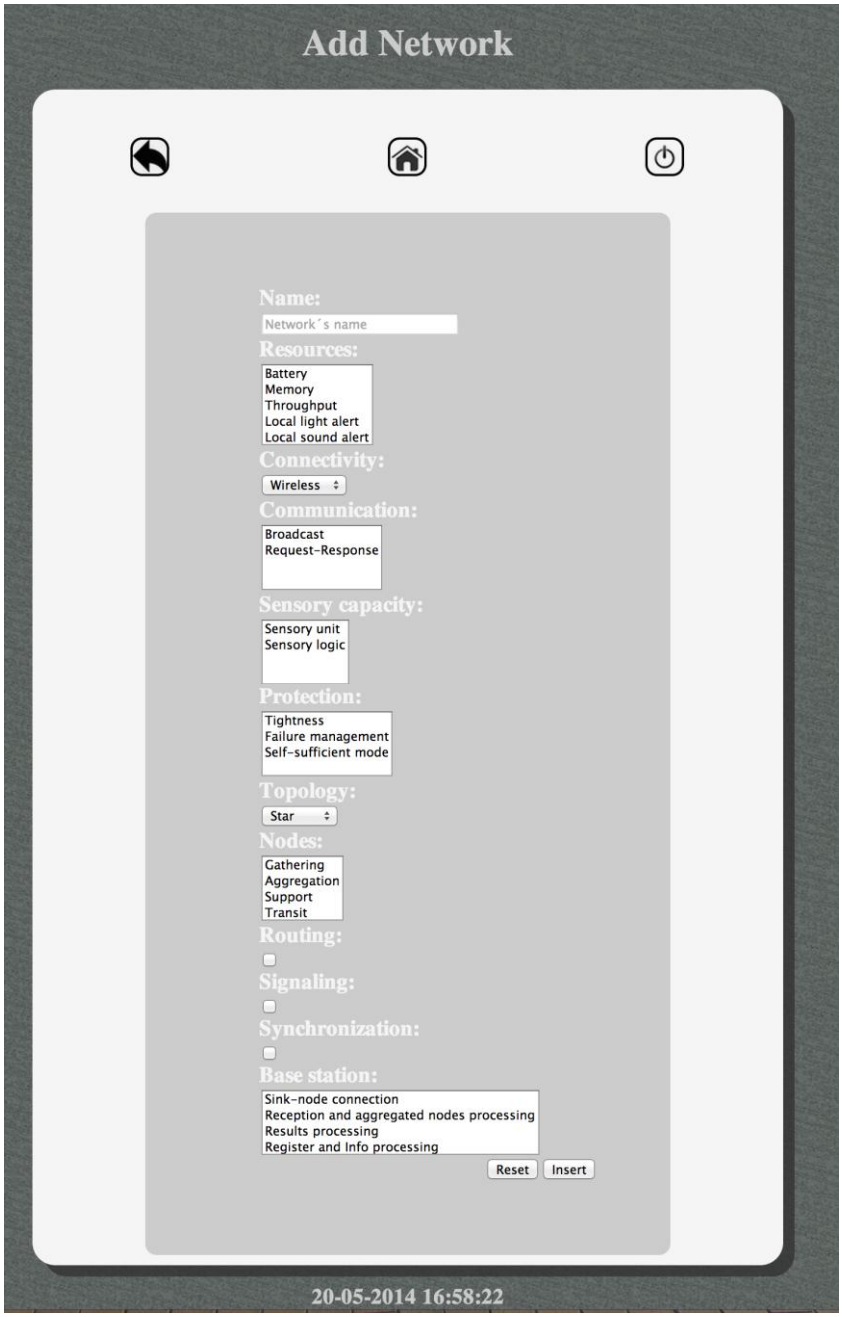


Figura 60. Añadir red

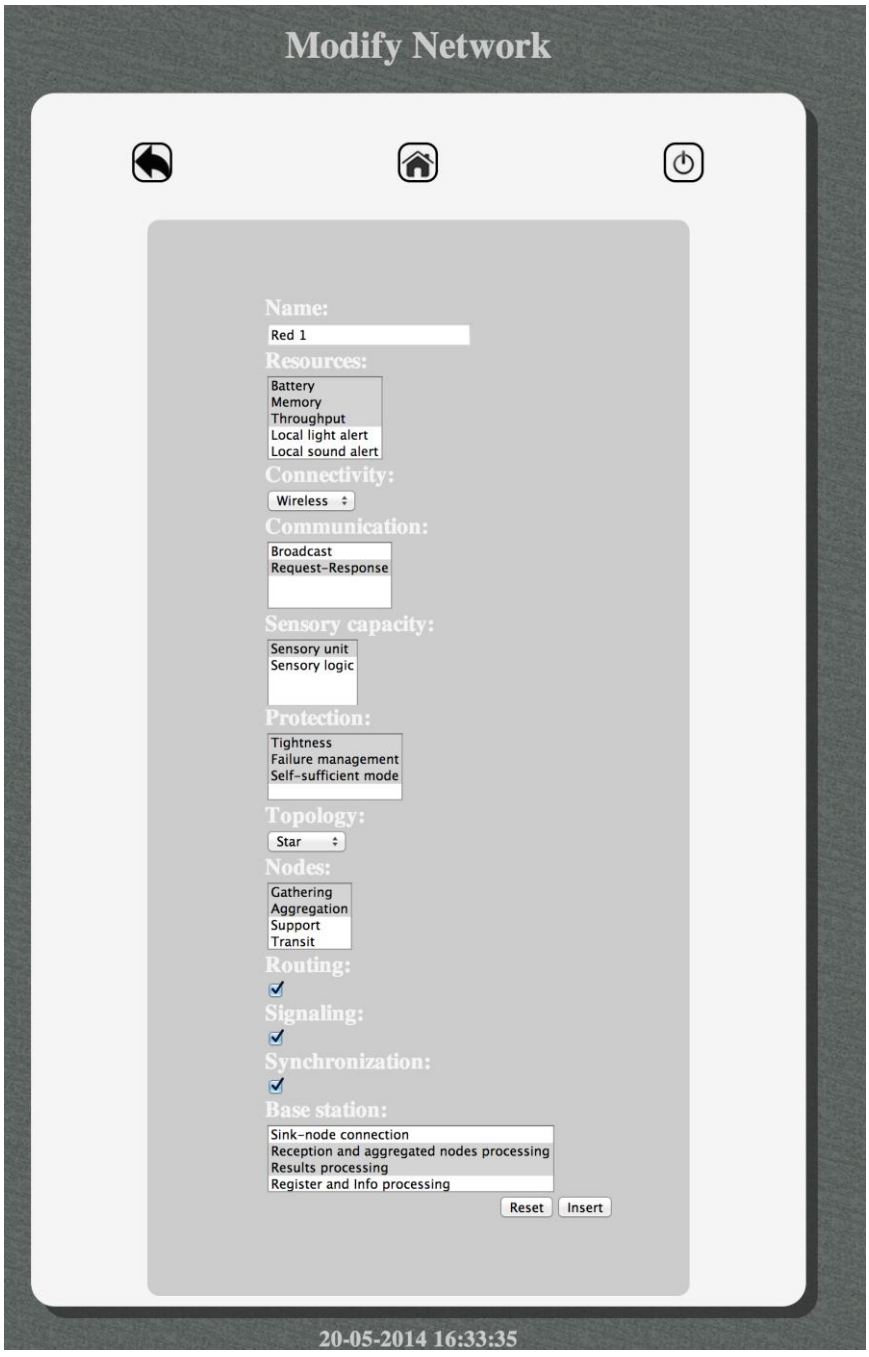


Figura 61. Modificar red

Es necesario mencionar que las características de la red son optativas, es decir, si en algún campo no se introduce alguna característica, en la base de conocimientos aparecerá vacío el campo correspondiente en la tabla.

7.4 Modificar Servicios

Aparte de todas las posibilidades que se ofrecen en "Inspeccionar Servicios", también se pueden realizar las mismas acciones que en las tres vistas anteriores con el mismo procedimiento.

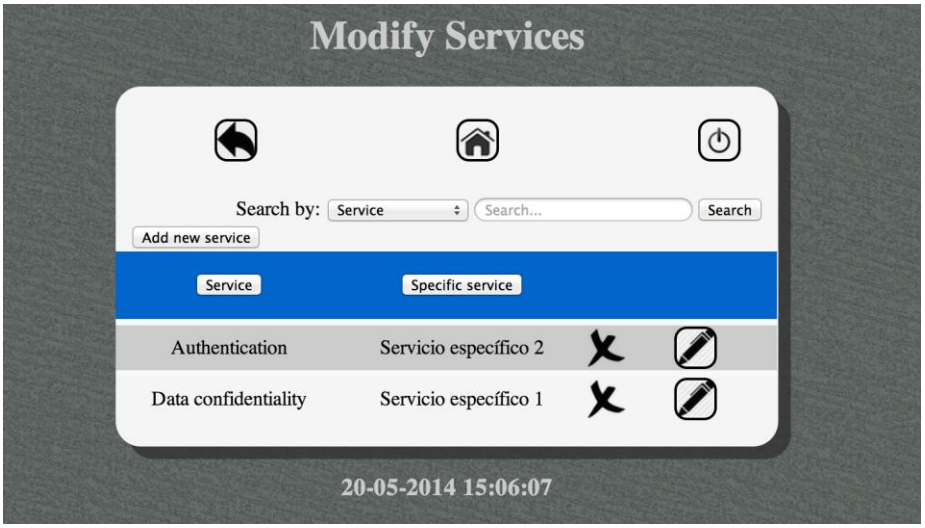


Figura 62. Modificar servicios

Las vistas 25 y 26 corresponden a la vista de añadir y de modificar un servicio. En ambas situaciones, los dos campos deben de ser rellenados obligatoriamente para tener éxito al añadir o al modificar.

Figura 63. Añadir servicio

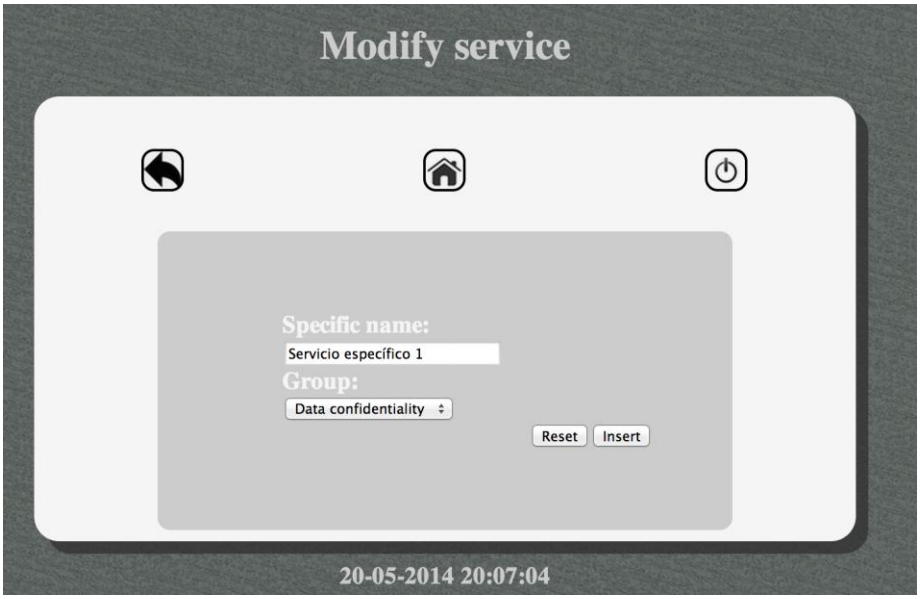


Figura 64 Modificar servicio

8 Base de hechos

Existen dos únicas formas de llegar a la vista que se observa en la Figura 65: a través de la evaluación por fichero o a través de la evaluación rápida. En ella se puede ver por una parte información relativa a la red que se ha introducido en la entrada y por otra los imperativos y los datos a los que se aplican que ha indicado previamente el usuario.

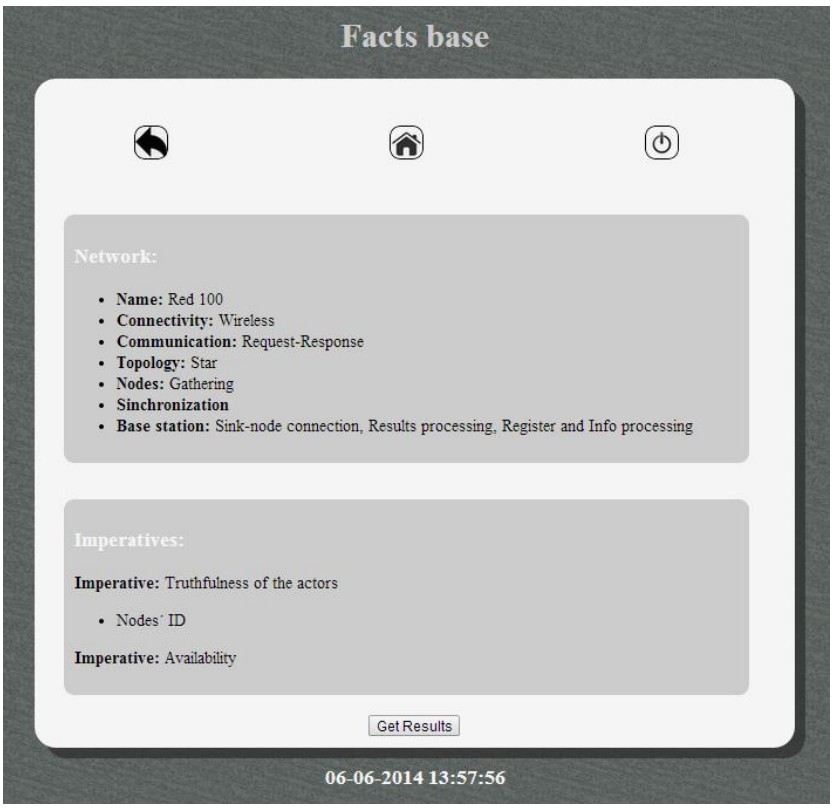


Figura 65. Base de hechos

9 Resultados

Una vez confirmada la información de la base de hechos, que es la que se utilizará en la lógica del programa, se procede a la obtención de resultados. Éstos se muestran en una vista como la de la Figura 66.

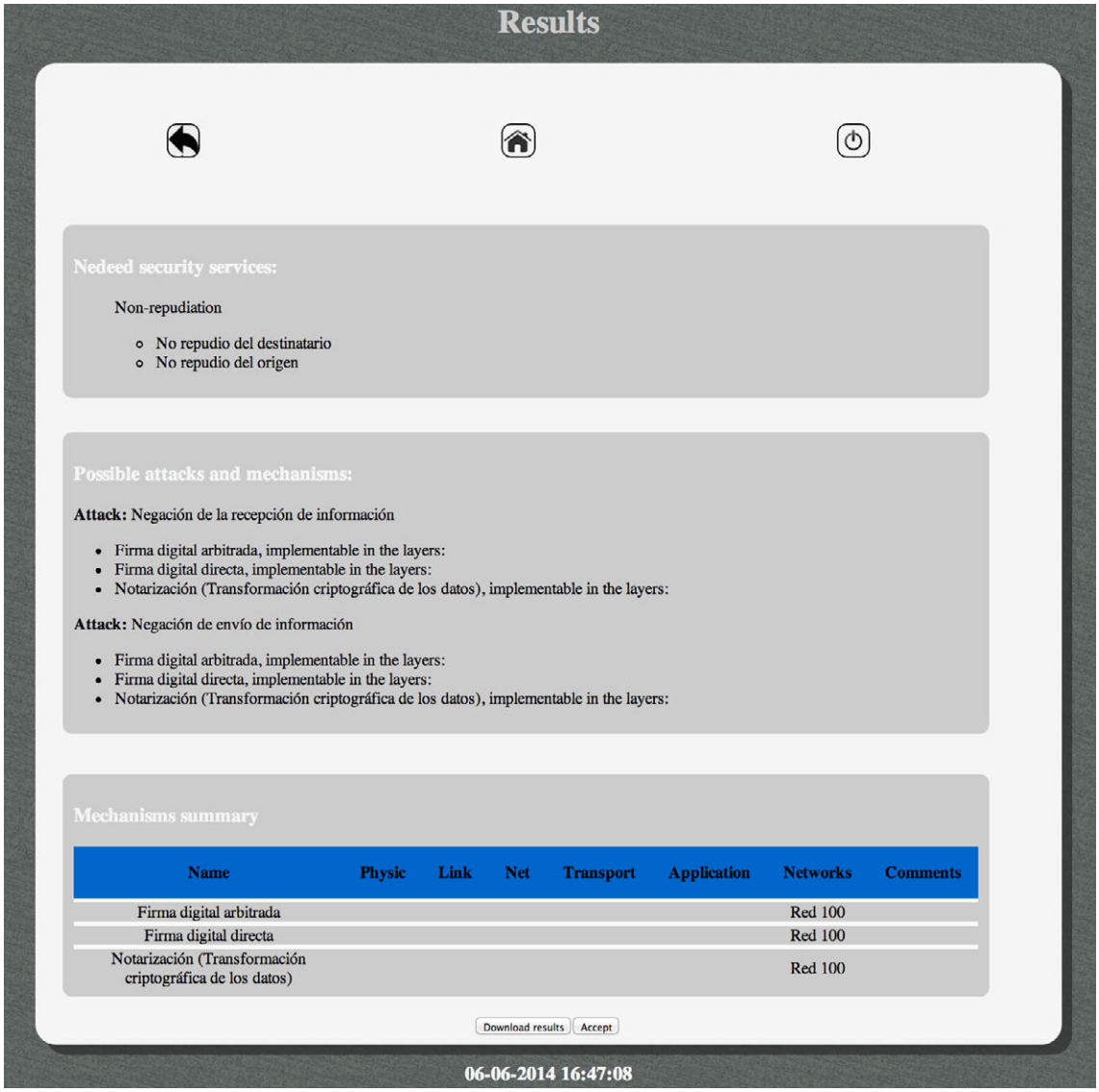


Figura 66. Resultados

Se pueden distinguir tres apartados:

- En el primer apartado, se especifican tanto los servicios generales como los servicios de seguridad particulares que se han de implementar en la red.
- En el segundo se puede visualizar la información de los ataques que pueden afectar a la red y de los mecanismos de seguridad que hay en la base de conocimientos que se pueden implementar para paliar ese ataque.
- Por último se hace un resumen con todos los mecanismos de seguridad que se pueden implementar en la red para asegurar los servicios del primero de los apartados.

También se ofrece la posibilidad de descargar un archivo con formato “txt” en el que se refleja un resumen de la información obtenida por pantalla.

10 Revisar/Descargar log

Esta es otra de las vistas que únicamente se ofrecen para el grupo de usuarios “administradores”. Dicha vista ofrece la posibilidad de revisar directamente las entradas del log que están guardadas en el fichero “log.txt” y también permite al usuario, haciendo click en el botón “Download File”, descargar del servidor dicho fichero en formato “txt”.

Figura 67. Revisar/Descargar log

11 ¿Quiénes somos?

Esta vista ofrece, a cualquier usuario, información acerca de la aplicación. Como se observa en la vista 28, hay una imagen del logo de CITSEM que al hacer click en ella abrirá una nueva pestaña en el navegador con la página web de dicho organismo.

Figura 68. ¿Quiénes somos?

12 Vistas de error

Durante la ejecución del programa, se pueden producir determinados fallos que se han de avisar al usuario para que así pueda actuar correctamente dependiendo del error que se haya producido.

Para ello se ofrecen cuatro vistas diferentes que determinarán el tipo de error que se ha producido y también, si es posible, la forma de actuación a partir de ese momento.

12.1 Error de autenticación

Esta página de error tiene el aspecto de la vista 29 y se produce cuando un usuario intenta hacer el log in en el sistema y ha olvidado su contraseña o su nombre de usuario.

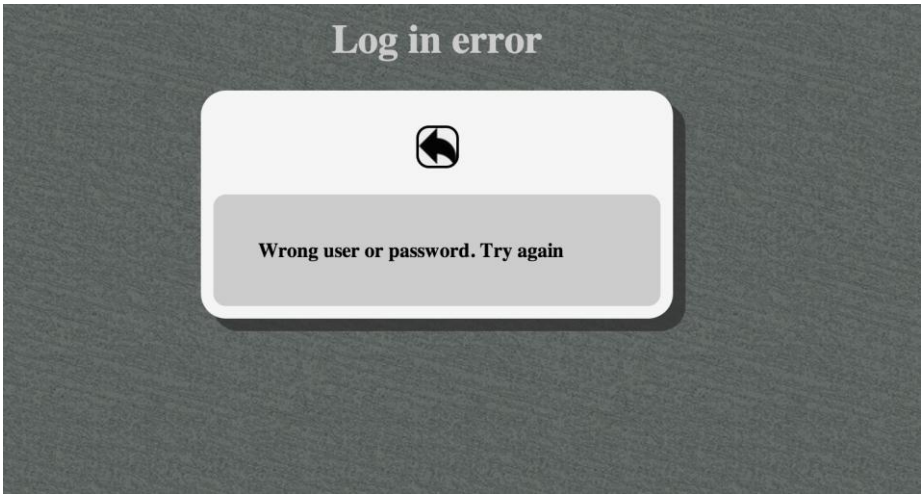


Figura 69. Log in Error

Como se puede ver en la vista 29. Sólo hay un botón que sirve para volver al punto de partida de la aplicación.

12.2 Error interno del servidor

Son errores que se producen porque por alguna razón el servidor no puede procesar la petición, ya sea por errores de ejecución, errores en la creación de los managed beans, etc.

Generalmente para esto es necesario que un técnico visualice el porqué se ha producido para poder evitarlo en un futuro. Otras veces, como se indica en la vista 30, bastaría con reiniciar el servidor. Por lo tanto son acciones que no dependen del usuario en sí.

Figura 70. Error interno del servidor

12.3 Error de seguridad

El error de seguridad es el que aparecerá a los usuarios con un determinado rol que no tengan acceso a una zona del programa.

Figura 71. Error de seguridad

La única posibilidad que se le ofrecerá a dicho usuario es el volver al menú principal haciendo click en el botón home ya que al menú principal tienen acceso todos aquellos usuarios que estén dados de alta en el sistema.

12.4 Error general

Se puede llegar a esta vista en el caso de que haya algún error en la lógica del programa. En caso de que aparezca este error, se ofrecerá una vista donde únicamente hay un botón que dirige al menú principal y se recomienda chequear el log para saber cuál ha sido el error exacto.

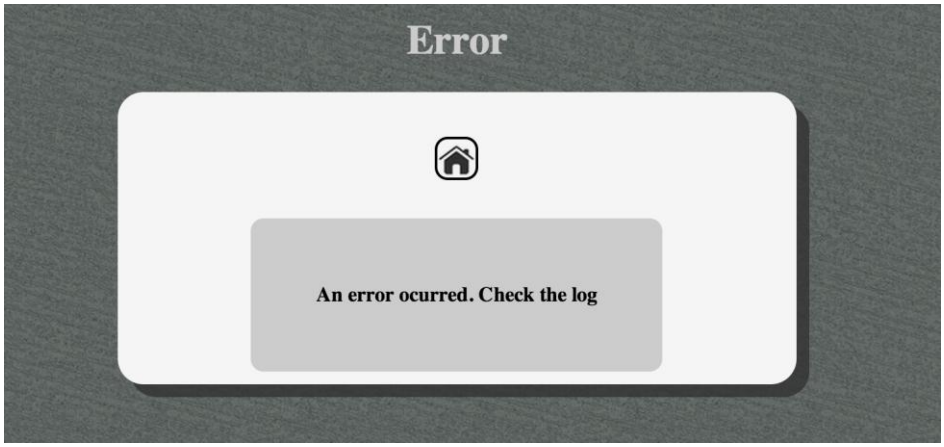


Figura 72. Error general

ANEXO II – BASE DE CONOCIMIENTOS EN EL CASO PRÁCTICO

Base de conocimientos para redes del tipo 2XX

Layer	Name	Security Service	Security mechanism	Network	Comments
2	COLLISION	AVAILABILITY INTEGRITY	Error correcting code Time diversity Limiting the rate of MAC requests Small frames S-MAC defensive method L-MAC defensive method B-MAC defensive method Identity protection Link Layer encryption	Red 200	[1], [2]
3	HELLO FLOOD		Bidirectional Verification Multi-path multi-base station routing	Red 200, Red 201	[3]
3	IMPERSONATION	AVAILABILITY INTEGRITY CONFIDENTIALITY AUTHENTICITY	Authentication Encipherment Secure routing Authentication with TTP Authentication Validation techniques Identity protection Link layer encryption Limiting the rate of MAC request Small frames	Red 200, Red 201	[2]
3	MESSAGE REPLAY		TinySec SNEP & TESLA	Red 200, Red 201	[3]
3	NEGLECT AND GREED	AVAILABILITY	Redundancy Probing	Red 200	[1]
1	NODE OUTAGE	AVAILABILITY INTEGRITY	Alternative path Robust protocols **Mechanism against node capture**	Red 200	[2]
2	RESOURCE EXAUSTION	AVAILABILITY	Limiting MAC admission control Rate Ramdom back-offs Time-Division multiplexing responses Protection of WSN ID and other information.	Red 200	[1], [2]
3	RUSHING	INTEGRITY	Table-guided protocols	Red 200	[4]
3	SPOOFING	INTEGRITY AUTHENTICITY	Communication Security TIK Random Key Predistribution TinySec SNEP & TESLA	Red 200	[2], [3]

			Alternative path Authentication Link Layer encryption Global shared key techniques		
3	TRAFFIC MANITULATION	AVAILABILITY INTEGRITY	**Traffic analysis defenses** **Collision attack defenses** **Unfairness attack defenses** **Misbehavior detection techniques** Identity protection Link layer encryption Limiting the rate of MAC requests Small frames	Red 200	[2]
1	EAVESDROPPING	CONFIDENTIALITY	Authentication exchange Authentication Access control Reduction in sensed data details Distributed processing Access restriction Encipherment	Red 200	[2]
1	JAMMING	AVAILABILITY INTEGRITY	Spread-spectrum Priority messages Lower duty cycle Region mapping JAM **Wormhole defenses** Limiting the rate of MAC requests Small frames S-MAC defensive method L-MAC defensive method B-MAC defensive method Identity protection Link Layer encryption	Red 200	[1]; [2], [3]
1	TAMPERING	INTEGRITY	Tamper-proof Hiding	Red 200	[1]
2	SYBIL		Radio Resource Testing Random Key Pre-distribution	Red 200	[3]
2	TRAFFIC ANALYSIS	CONFIDENTIALITY	Asymetric encipherment Dynamic Routing control (security labels) Symetric encipherment Traffic padding	Red 200	[1]
2	UNFAIRNESS	AVAILABILITY INTEGRITY	Small frames	Red 200	[1], [2]
3	BLACKHOLE	INTEGRITY	Authorization, Monitoring	Red 200	[1], [3]

			Redundancy REWARD		
3	FLOODING	AUTHENTICATION AVAILABILITY	Client Puzzles Increasing bandwidth	Red 200	[1]
3	HOMMING	CONFIDENTIALITY	Encipherment	Red 200	[1]
3	INFORMATION SPOOFING		Statistical En-Route Filtering	Red 200	[3]
3	MISDIRECTION	INTEGRITY	Egress filtering Authorization Monitoring Redundancy	Red 200	[01]
3	SELECTIVE MESSAGE FORWARDING	AVAILABILITY	Multiple disjoint routing paths Diversity coding	Red 200	[3]
3	SINKHOLE	AVAILABILITY INTEGRITY AUTHENTICITY	Detection on MintRoute Geographical routing protocols Learning global map Probabilistic next hop selection leveraging global knowledge Verifying information advertised of neighbor nodes Authentication Link Layer encryption Global shared key techniques Routing access restriction **Wormhole defenses** Key management Secure routing	Red 200, Red 201	[2]
3	WORMHOLE	CONFIDENTIALITY AUTHENTICITY	TIK Packet leach/leashes techniques MAD and OLSR protocols Directional antennas Multi-dimensional scaling algorithm (scalability) Local neighborhood information DAWWSEN protocol Clustering-based and geographical routing protocols leveraging global knowledge Verifying information announced of neighbor nodes Graphical Position System Ultrasound Global clock synchronization Combinational methods (such as radiowaves and ultrasound) Authentication	Red 200	[2], [3]

			Link Layer encryption Global shared key techniques		
4	DESYNCHRONIZATION	AVAILABILITY AUTHENTICITY	Authentication Time synchronization Maintaining proper timing	Red 200	[1], [2]
7	CLOCK SKEWING	INTEGRITY	Supervising timing information in synchronization packets Beacon packets are broadcasted Syncronized by the access point periodically	Red 200	[5]
7	REPUDIATION	NON-REPUDIATION	Authentication with TTP	Red 200	[6]

Referencias del Anexo II

[1] P. J. Walters, Z. Liang, W. Shi y V. Chaudhary, «Wireless Sensor Network Security: A Survey,» *IEEE Commun. Surveys & Tutorials,* pp. 52-73, 2009.

[2] Dr. Shahriar Mohammadi and Hossein Jadidoleslamy, «A Comparison Of Link Layer Attacks On Wireless Sensor Networks, » *International journal on applications of graph theory in wireless ad hoc networks and sensor networks*, (GRAPH-HOC) Vol.3, No.1, Marzo 2011.

[3] Al-Sakib Khan Pathan, Kyung Hee, Hyung-Woo Lee, «Security in Wireless Sensor Networks: Issues and Challenges, », Febrero. 20-22.

[4] J. A. Bertolín, Análisis de riesgos y contramedidas en REDES MANET, Seguridad en redes, REE, Febrero 2012.

[5] Kai Xing, Shyaam Sundhar Rajamadam Srinivasan, Manny Rivera, «Attacks and Countermeasures in Sensor Networks: A Survey, », Springer 2005.

[6] S. Gupta, H. K. Verma y A. L. Sangal, «Security Attacks & Prerequisite for Wireless Sensor Networks,» *International Journal of Engineeering and Advanced Technology (IJEAT),* vol. 2, pp. 558-566, 2013.