



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Trabajo de
Desarrollo

EVALUADOR DE CABECERAS HTTP PARA SERVICIOS VÍA WEB

**GRADO EN INGENIERÍA INFORMÁTICA DEL
SOFTWARE**

TRABAJO DE FIN DE GRADO

AUTOR

Daniel Villanueva Pérez

TUTOR

José Manuel Redondo López

Junio 2022

Este documento ha sido creado basándose en la plantilla elaborada por JOSÉ MANUEL REDONDO LÓPEZ. [1][2]

Declaración Responsable

El alumno: Daniel Villanueva Pérez

Con DNI: 71732399E

Y UO: 251399

DECLARA

Que esta obra es completamente original y se han citado debidamente las fuentes utilizadas durante la realización de esta.

Y para que conste, lo firma en Oviedo, a 6 de junio de 2022

Firmado:

Agradecimientos

Me gustaría dedicar este espacio a todas las personas que, de una forma u otra, han hecho posible la realización de este TFG:

Primero de todo, quiero dar las gracias a mi familia, que siempre ha estado interesada en todo lo que he hecho y, a pesar de que todavía no tienen muy claro a qué dedico tantas horas delante del ordenador, se han molestado en leer este documento y sugerirme correcciones. Pero sobre todo, quiero agradecerles su amor incondicional y que hayan estado ahí siempre que los he necesitado. Este trabajo, la culminación de mi carrera universitaria, jamás habría sucedido sin todas las cosas, pequeñas y grandes, que han hecho por mi durante todos estos años.

También quiero dar las gracias a Bego, mi pareja, por el abrumador apoyo que me ha dado en todo momento. No importa lo cansado, triste o derrotado que pueda haberme sentido a veces, ella siempre me ha dado fuerzas y motivación para seguir adelante y completar este grado (y este trabajo).

De igual forma, quiero agradecer a mis amigos por haber estado conmigo en los mejores y peores momentos de mi paso por la universidad. Eduardo, Tomás, Daniel y Tamara, siempre dispuestos a escucharme, aconsejarme, o reconfortarme cuando lo he necesitado. Y Juan, que siempre confió en mi incluso cuando ni yo mismo lo hacía, su constante interés en el progreso del proyecto no ha pasado desapercibido.

Y por supuesto, quiero agradecer a todos los profesores de la Escuela de Ingeniería Informática que me han dado clase, en especial a Alberto Manuel Fernández, Darío Álvarez, Noelia Rico, Fernando Álvarez y, sobre todo, José Manuel Redondo, para mí un profesor ejemplar, continuamente interesado en el aprendizaje, crecimiento y bienestar de sus estudiantes; y tutor de este proyecto, cuya supervisión e increíble involucración en el mismo han sido clave para llegar a buen puerto.

Finalmente, me gustaría agradecer a mi yo del año 2016 por no haberse rendido, pese a tener motivos de sobra para hacerlo. No me puedo ni imaginar cómo sería mi vida ahora mismo de no haber sacado fuerzas de flaqueza en aquel momento tan oscuro, pero tengo la certeza de que ninguna otra realidad alternativa podría igualar esta situación de felicidad, bienestar, y perspectivas de futuro en la que me encuentro en este momento.

Hay una parte de todos vosotros en cada clase a la que he asistido, cada bloque de código que he desarrollado, cada página de apuntes que he subrayado, cada entrega pendiente que me ha desvelado, y cada examen que he superado. En mayor o menor medida, este trabajo, esta carrera, es vuestra también.

Gracias por haber estado ahí siempre.

Índice de contenido

Capítulo 1. ¿Qué es este Trabajo?	13
1.1 Resumen	13
1.2 Palabras clave	13
1.3 Abstract	14
1.4 Keywords	14
Capítulo 2. Planificación del Sistema de Información	15
2.1 PSI 1: Inicio del Plan de Sistemas de Información	16
2.1.1 PSI 1.1: Análisis de la Necesidad del PSI	16
2.1.2 PSI 1.2: Identificación del Alcance del PSI	16
2.2 PSI 2: Definición y Organización del PSI	17
2.2.1 PSI 2.1: Especificación del Ámbito y Alcance	17
2.3 PSI 3: Estudio de la Información Relevante	18
2.3.1 PSI 3.1: Conceptos Teóricos	18
Capítulo 3. PSI 4: Definición de la Arquitectura Tecnológica	21
3.1 PSI 4.1: Identificación de las Necesidades de Infraestructura Tecnológica	22
Capítulo 4. Planificación y Gestión del TFG	23
4.1 Planificación del proyecto	24
4.1.1 Identificación de Interesados	24
4.1.2 PBS y OBS	24
4.1.3 Planificación Inicial. WBS	25
4.1.4 Riesgos	29
4.1.5 Presupuesto Inicial	31
4.2 Ejecución del Proyecto	32
4.2.1 Plan de Seguimiento de la Planificación	32
4.2.2 Bitácora de Incidencias del Proyecto	32
4.2.3 Riesgos	33
4.3 Cierre del Proyecto	34
4.3.1 Planificación Final	34
4.3.2 Informe Final de Riesgos	36

4.3.3	Presupuesto Final de Costes	36
4.3.4	Informe de Lecciones Aprendidas	36
Capítulo 5.	Análisis del Sistema de Información	37
5.1	ASI 1: Definición del Sistema	38
5.1.1	Determinación del Alcance del Sistema	38
5.2	ASI 2: Establecimiento de Requisitos.....	39
5.2.1	Obtención de los Requisitos del Sistema	39
5.2.2	Identificación de Actores del Sistema	41
5.2.3	Especificación de Casos de Uso	42
5.3	ASI 3: Identificación de Subsistemas de Análisis	44
5.3.1	Descripción de los Subsistemas.....	44
5.3.2	Descripción de los Interfaces entre Subsistemas.....	44
5.4	ASI 4: Análisis de los Casos de Uso	45
5.4.1	Caso de Uso 1 – Escanear Red.....	45
5.4.2	Caso de Uso 2 – Comparar Informes	46
5.5	ASI 5: Análisis de Clases	47
5.5.1	Diagrama de Clases.....	47
5.5.2	Descripción de las Clases	47
5.6	ASI 6: Definición de Interfaces de Usuario.....	48
5.6.1	Descripción de la Interfaz.....	48
5.6.2	Definición del aspecto de la Interfaz	49
5.6.3	Descripción del Comportamiento de la Interfaz	50
5.7	ASI 7: Especificación del Plan de Pruebas	51
5.7.1	Pruebas Unitarias.....	51
5.7.2	Pruebas de Integración	52
Capítulo 6.	Diseño del Sistema de Información.....	53
6.1	DSI 1: Diseño de Casos de Uso Reales	54
6.1.1	Caso de Uso 1 – Escanear Red.....	54
6.1.2	Caso de Uso 2 – Comparar Informes	55
6.2	DSI 2: Diseño de Clases	56
6.2.1	Diagrama de Clases.....	56
6.2.2	Descripción de las Clases	56

6.3	DSI 3: Diseño de la Arquitectura de Módulos del Sistema	58
6.3.1	DSI 3.1 Diagrama de Componentes del Sistema	58
6.3.2	DSI 3.2 Revisión de la Interfaz de Usuario	59
6.4	DSI 4: Especificación Técnica del Plan de Pruebas	62
6.4.1	Pruebas Unitarias.....	62
6.4.2	Pruebas de Integración	66
Capítulo 7.	Construcción del Sistema de Información	69
7.1	CSI 1: Preparación del Entorno de Generación y Construcción	70
7.1.1	Lenguajes de programación	70
7.1.2	Herramientas y programas usados para el desarrollo	70
7.2	CSI 2: Ejecución de las Pruebas Unitarias	71
7.3	CSI 3: Ejecución de las Pruebas de Integración	73
7.4	CSI 4: Elaboración de los Manuales de Usuario	75
7.4.1	Manual de Instalación.....	75
7.4.2	Manual de Ejecución.....	75
7.4.3	Manual de Usuario	76
7.4.4	Manual del Programador	80
Capítulo 8.	Conclusiones y Ampliaciones	81
8.1	Conclusiones.....	82
8.2	Ampliaciones	83
Apéndices	85	
Plan de Gestión de Riesgos.....	86	
Matriz de probabilidad e impacto	86	
Estructura de Desglose de Riesgo. RBS.....	87	
Referencias bibliográficas.....	88	
Contenido entregado en los anexos	91	
Contenidos	91	

Índice de Figuras

PRODUCT BREAKDOWN STRUCTURE (PBS) DEL PROYECTO	24
VISTA GENERAL DE LA PLANIFICACIÓN DEL PROYECTO	25
PLANIFICACIÓN DEL PROYECTO PARA LOS CAPÍTULO 1-4 DE LA DOCUMENTACIÓN	26
PLANIFICACIÓN DEL PROYECTO PARA EL CAPÍTULO 5 DE LA DOCUMENTACIÓN	26
PLANIFICACIÓN DEL PROYECTO PARA EL CAPÍTULO 6 DE LA DOCUMENTACIÓN	27
PLANIFICACIÓN DEL PROYECTO PARA EL DESARROLLO DE LA APLICACIÓN Y DE LAS PRUEBAS	27
PLANIFICACIÓN DEL PROYECTO PARA EL CAPÍTULO 7 DE LA DOCUMENTACIÓN, EL CIERRE DEL PROYECTO, Y EL CAPÍTULO 8.....	28
DURACIÓN Y FECHA REALES VS PLANIFICADAS DE LAS TAREAS DE LOS CAPÍTULO 1-4 DE LA DOCUMENTACIÓN (LÍNEA BASE A MEDIO PROYECTO).....	34
DURACIÓN Y FECHA REALES VS PLANIFICADAS DE LAS TAREAS DE LOS CAPÍTULO 5-6 DE LA DOCUMENTACIÓN (LÍNEA BASE A MEDIO PROYECTO).....	35
DURACIÓN Y FECHA REALES VS PLANIFICADAS DE LAS TAREAS DE CONSTRUCCIÓN DE LA APLICACIÓN Y LOS CAPÍTULO RENTANTES DE LA DOCUMENTACIÓN	35
CASO DE USO 1 – ESCANEAR RED	42
CASO DE USO 2 – COMPARAR INFORMES	43
DESCRIPCIÓN DEL “CASO DE USO 1 – ESCANEAR RED” CON UN DIAGRAMA DE ROBUSTEZ (I).....	45
DESCRIPCIÓN DEL “CASO DE USO 2 – COMPARAR INFORMES” CON UN DIAGRAMA DE ROBUSTEZ (II)	46
DIAGRAMA DE CLASES DEL PROYECTO (FASE DE ANÁLISIS)	47
PROTOTIPO DE LA INTERFAZ DEL ESCÁNER DE RED	49
PROTOTIPO DE LA INTERFAZ DEL COMPARADOR DE INFORMES.....	49
DIAGRAMA DE SECUENCIA DEL CASO DE USO 1 – ESCANEAR RED.....	54
DIAGRAMA DEL CASO DE USO 2 – COMPARAR INFORMES	55
DIAGRAMA DE CLASES DEL PROYECTO (FASE DE DISEÑO)	56
DIAGRAMA DE COMPONENTES DEL SISTEMA	58
TRAZA DE EJECUCIÓN DE LOS TEST UNITARIOS DEL ESCÁNER DE RED	72
TRAZA DE EJECUCIÓN DE LOS TEST UNITARIOS DEL ESCÁNER DE PUERTOS	72
TRAZA DE EJECUCIÓN DE LOS TEST UNITARIOS DEL COMPARADOR DE INFORMES	72
TRAZA DE EJECUCIÓN DE LOS TEST DE INTEGRACIÓN DE LA HERRAMIENTA	74
FRAGMENTO DE UN INFORME GENERADO POR LA HERRAMIENTA, EN EL QUE FIGURAN CVES ENCONTRADAS EN UN SERVIDOR DE PRUEBA DEL PROYECTO	84

Índice de Tablas

DESGLOSE DEL CÁLCULO DE LA PRIORIDAD DE LOS DISTINTOS RIESGOS DEL PROYECTO	30
ESTRATEGIAS A APLICAR PARA MINIMIZAR LOS RIESGOS DEL PROYECTO	31
MATRIZ DE PROBABILIDAD E IMPACTO DE RIESGOS DEL PROYECTO	86
ESTRUCTURA GENERAL DEL FICHERO ANEXO ENTREGADO	91
ESTRUCTURA DE LA CARPETA “DESARROLLO” DEL FICHERO ANEXO ENTREGADO	91

Capítulo 1. ¿QUÉ ES ESTE TRABAJO?

1.1 RESUMEN

Este proyecto consiste en una aplicación de escritorio que sirve para automatizar el análisis de cabeceras HTTP en servicios web desde el punto de vista de la seguridad. Ya que, aunque los servicios web sean algo común hoy en día, no siempre se implementan correctamente y por ello pueden ser objetivo de ataques ‘man-in-the-middle’, ‘click-jacking’, ‘cross-site-scripting’, u otros, haciendo que la integridad de los sistemas informáticos se puede ver comprometida, con todos los riesgos que eso supone. Realizar una auditoría de seguridad de dichos servicios sin autorización de sus responsables no es legal, pero sí que se puede obtener una estimación del nivel de seguridad de cualquier servicio examinando las cabeceras HTTP incluidas en las respuestas a las peticiones que se le hagan.

Gracias a ese principio, la herramienta desarrollada permite que, tanto profesionales de la ciberseguridad como amateurs, puedan examinar de forma fácil, rápida, intuitiva y no intrusiva el estado de los servicios web de las máquinas de una red. Los resultados de los análisis se guardan en informes inteligibles, que incluyen los problemas encontrados y recomendaciones de actuación para los mismos, y que además se pueden comparar entre sí de forma automática para observar la evolución en el tiempo de uno o más servicios web.

Aunque el trabajo no tiene un cliente real como tal, sí que responde a las necesidades que algunos miembros de la comunidad universitaria han comunicado a José Manuel Redondo, tutor del proyecto. Se espera que la aplicación desarrollada agilice la localización y solución de problemas en los servidores de la red de la Universidad de Oviedo.

1.2 PALABRAS CLAVE

Cabecera HTTP, Protocolo de Transferencia de Hipertexto, servicio web, ciberseguridad, análisis



1.3 ABSTRACT

This project consists in a desktop application that helps to automate the analysis of HTTP headers in web services from the point of view of security. Even though web services are common nowadays, they are not always implemented correctly and thus they can be the object of attacks, like ‘man-in-the-middle’, ‘click-jacking’, or ‘cross-site-scripting’, among others, making that the integrity of computer systems can be compromised, with all the risks that it entails. To perform a security audit of those services without previous authorization of their responsible is not legal, but it is possible to obtain an estimation of the security level of any service by examining the HTTP headers that are included in the responses to the requests made to them.

Thanks to that principle, the developed tool allows to either cybersecurity professionals as well as amateurs to examine easily, fast, intuitively, and non-intrusively the status of the web services of machines in a network. The results of the analysis are saved in intelligible reports, that include the found problems and recommendations to solve them, besides allowing to make comparisons between them automatically in order to see the evolution in time of web services.

Even though the work does not have a real client, it responds to the needs that some members of the university community have communicated to José Manuel Redondo, tutor of the project. It’s expected that the developed tool will speed up the location and problem solving of servers in the network of the University of Oviedo.

1.4 KEYWORDS

HTTP Header, Hypertext Transfer Protocol, web service, cybersecurity, analysis

Capítulo 2. PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI

2.1 PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

2.1.1 PSI 1.1: Análisis de la Necesidad del PSI

El objetivo del proyecto es desarrollar una aplicación de escritorio que, dada una red de máquinas, pueda encontrar todos los servicios web que ofrecen éstas y analice qué cabeceras HTTP se encuentran implementadas o no en cada uno de ellos. Esta información después tiene que sintetizarse en uno o más informes que incluyan todos los detalles del análisis de forma comprensible, junto con recomendaciones para resolver los problemas que haya encontrado. Además, estos informes tienen que poder compararse entre sí, de forma que se pueda ver la evolución de un servicio web entre dos análisis en momentos distintos. Finalmente, estas funcionalidades tienen que ser accesibles desde una interfaz de usuario.

2.1.2 PSI 1.2: Identificación del Alcance del PSI

Actualmente, en la red de servidores de la Universidad de Oviedo, hay una serie de máquinas faltas de mantenimiento, obsoletas, o directamente olvidadas en las que se ofrecen servicios web inseguros y que pueden suponer un problema para toda la organización. Muchos de estos, al no ser accesibles desde internet, no se pueden examinar con las herramientas habituales (como Shodan [3], por ejemplo), haciendo que analizarlos (e incluso, encontrarlos), para reclamar a sus responsables que tienen que encargarse de ellas de alguna forma, sea un procedimiento complicado y laborioso.

Si alguien quisiera llevar a cabo este proceso sin la herramienta desarrollada, tendría que usar una aplicación (como nmap [4], por ejemplo) para encontrar las máquinas en la red y escanear sus puertos, filtrar los que hospeden servicios web, mandar peticiones HTTP a cada uno de ellos, revisar una por una las respuestas que reciba, y finalmente escribir los resultados de este trabajo en un documento.

Por tanto, los objetivos estratégicos a lograr para considerar el proyecto exitoso son:

- Automatizar la detección de máquinas en una red.
- Automatizar el escaneo de servicios web en una máquina.
- Automatizar la generación de informes de los servicios web de una máquina.
- Automatizar la comparación de informes de los servicios web de una máquina.

2.2 PSI 2: DEFINICIÓN Y ORGANIZACIÓN DEL PSI

2.2.1 PSI 2.1: Especificación del Ámbito y Alcance

A partir de los objetivos estratégicos vistos en el apartado anterior, podemos dividir el proyecto en las siguientes fases, con sus respectivos objetivos en cada una de ellas:

Fase 1: Escaneo de redes y máquinas

Se desarrollará un módulo de la aplicación que pueda detectar qué máquinas hay en una red. Para ello, el usuario podrá escribir un rango de direcciones IP y la herramienta tendrá que determinar cuáles de ellas tienen un host asignado y cuáles no. Por cada host existente, la aplicación revisará cada uno de sus puertos y determinará cuáles (de haberlos) hospedan un servicio web. Después, mandará una petición HTTP a todos los servicios, y analizará sus respuestas para obtener qué cabeceras implementa cada uno de ellos.

Objetivos de la fase:

- Dado un rango de direcciones IP, poder encontrar cuáles tienen un host asignado y cuáles no.
- Dado un conjunto de hosts, determinar en qué puertos hay hospedados servicios web.
- Dado un servicio web, determinar qué cabeceras implementa a partir de una respuesta suya.

Fase 2: Construcción de informes

Se creará un módulo de la aplicación que sintetice la información encontrada en los análisis. Tendrá que generar un informe por cada host que tenga servicios web, y un informe adicional que incluya los resultados unidos de todos los hosts (si hubiese más de uno). Estos informes tendrán que incluir el estado de las cabeceras que se esperaba encontrar (las necesarias para considerar seguro al servicio web), y de las otras que implemente. Los problemas que encuentre estarán señalados y se tendrá que ofrecer información de referencia para resolverlos. Además, los nombres de los informes tendrán que incluir información significativa, como la IP de la máquina a la que pertenecen, el número de cabeceras esperadas que implementa y la fecha en la que se haya realizado el análisis.

Objetivos de la fase:

- Desarrollar una estructura de datos que incluya qué cabeceras HTTP debe implementar un servicio web (y qué valores deben incluir) para considerarlo seguro.
- A partir de los resultados de un análisis, automatizar la generación de los informes necesarios para representar la información encontrada de forma accesible a un usuario.

Fase 3: Comparación de informes

Se desarrollará un módulo de la aplicación que, a partir de dos informes generados por la herramienta, pueda encontrar las diferencias que haya entre ellos. Con esa información construirá un informe nuevo, en el que señale qué cosas se han añadido, modificado, o eliminado en el segundo con respecto al primero.

Objetivos de la fase:

- A partir de dos informes, construir un informe nuevo en el que se señale qué ha cambiado de uno con respecto a otro.

2.3 PSI 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE

2.3.1 PSI 3.1: Conceptos Teóricos

Servicio Web:

Los servicios web son aplicaciones que facilitan un servicio a través de la Web. Destacan por ser una tecnología distribuida: múltiples clientes pueden acceder a un solo servicio; y multiplataforma: cliente y servidor no han de tener la misma configuración para comunicarse [5]. Son el sujeto de análisis del proyecto.

HTTP:

HTTP son las siglas de *“Hypertext Transfer Protocol”*, el nombre del protocolo en el que se fundamentan los intercambios de datos en la Web. Es un protocolo que sigue la estructura cliente-servidor, es decir, los clientes (normalmente navegadores Web) realizan una petición de datos a un servidor, y este responde devolviendo los datos pedidos al cliente [6]. HTTPS, *“Hypertext Transfer Protocol Secure”*, es una evolución de este protocolo, en el que se cifra la información que se envía y recibe gracias a la tecnología SSL. Tanto las peticiones como las respuestas HTTP/HTTPS incluyen cabeceras [7], una serie de metadatos que incluyen información fundamental para realizar correctamente estas transacciones. Este trabajo usa las cabeceras obtenidas en las respuestas HTTP para obtener información significativa de los servicios web que las han mandado.

SSL:

SSL son las siglas de *“Secure Sockets Layer”*, una tecnología usada para realizar conexiones seguras en la Web. Se basa en utilizar algoritmos de cifrado para codificar los datos de forma que sean imposibles de leer por un tercero [8]. Esta codificación se realiza a partir de certificados digitales, archivos expedidos por entidades certificadoras que vinculan una clave criptográfica con los datos de una organización [9]. Este trabajo usa SSL para poder realizar conexiones con servicios web que lo implementen.



Puerto:

Un puerto es una abstracción software que puede entenderse como un punto de conexión a una máquina. El tipo de conexión que pueda realizarse en un puerto dependerá del protocolo que use éste, que determinará el número de clientes que puede escuchar y el formato de las peticiones y respuestas que se puedan realizar (por ejemplo, no es lo mismo habilitar un puerto para una conexión SSH que para una conexión FTP) [10]. En nuestro caso los únicos puertos que nos interesan son los que implementen el protocolo HTTP o HTTPS.

Socket:

Si un puerto representa por dónde se transporta la información de una conexión entre dos máquinas, un socket representa la conexión en sí [11]. La aplicación crea un socket por cada puerto que intenta escanear, para después descartarlo en los siguientes casos:

- Si el puerto no está activo.
- Si el puerto está activo pero no está hospedando un servicio web.
- Si el puerto está activo y hospeda un servicio web, pero la herramienta ya ha obtenido la información que necesitaba de él.

Capítulo 3. PSI 4:

DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

FASE DE PLANIFICACIÓN

PSI



3.1 PSI 4.1: IDENTIFICACIÓN DE LAS NECESIDADES DE INFRAESTRUCTURA TECNOLÓGICA

Como el proyecto consiste en una aplicación de escritorio, la infraestructura que necesita es bastante reducida. La herramienta se ha testado en una máquina virtual Ubuntu con 2GB de RAM, 64MB de memoria gráfica, y 64GB de espacio de almacenamiento, pero con menos recursos debería funcionar sin problemas. El sistema operativo de la máquina no es relevante, porque solo necesita tener instalado nmap [4] y un intérprete de Python 3.9 que tenga instalados los paquetes PySimpleGUI [13], requests [14] y python-nmap [12], ya que el resto de las dependencias están entre los paquetes por defecto de esa versión del intérprete. En caso de querer ejecutar los test, haría falta también instalar el paquete PyAutoGUI [15].

Capítulo 4. PLANIFICACIÓN Y GESTIÓN DEL TFG

FASE DE DESARROLLO

4.1 PLANIFICACIÓN DEL PROYECTO

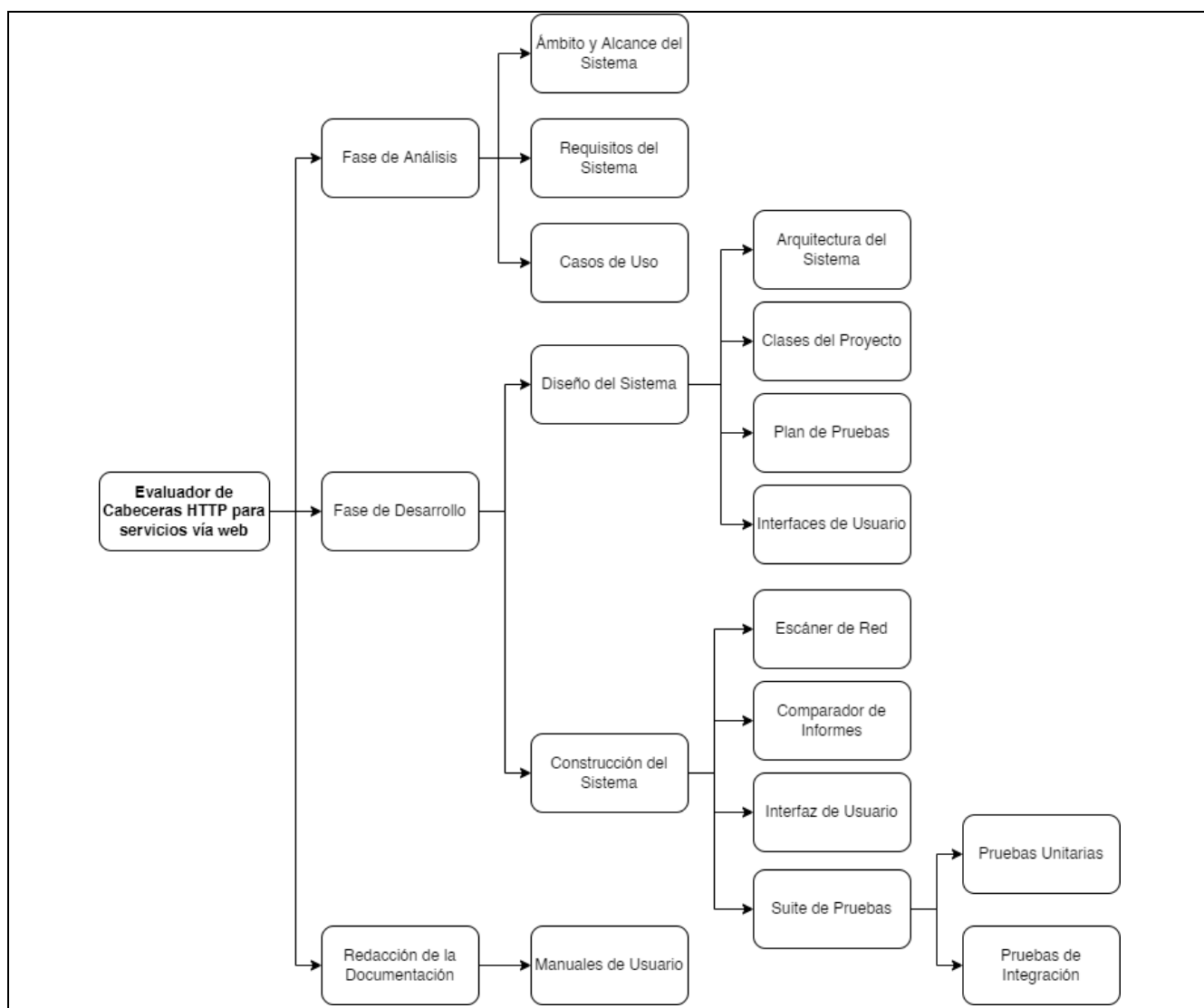
4.1.1 Identificación de Interesados

Los interesados identificados son los siguientes:

- José Manuel Redondo, tutor del proyecto
- Daniel Villanueva Pérez, analista y desarrollador del proyecto
- Miembros de la Universidad de Oviedo que tengan relación con servicios web internos
- Personas interesadas en ciberseguridad, en especial seguridad web

4.1.2 PBS y OBS

A continuación se muestra el diagrama del *Product Breakdown Structure (PBS)*:

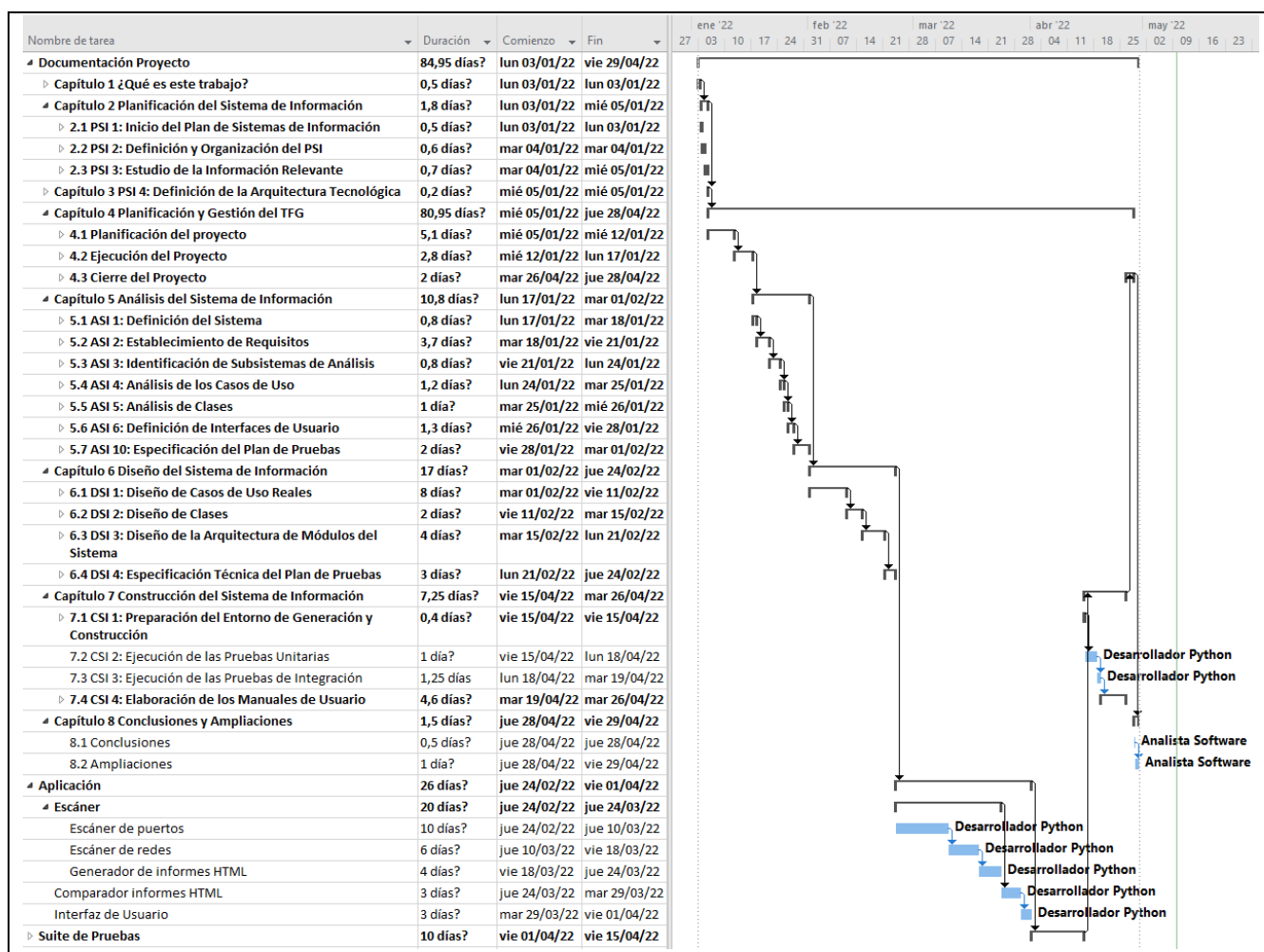


Product Breakdown Structure (PBS) del Proyecto

Respecto al diagrama del *Organizational Breakdown Structure (OBS)*, se ha omitido puesto que el único elemento que figuraría en él sería el autor del proyecto.

4.1.3 Planificación Inicial. WBS

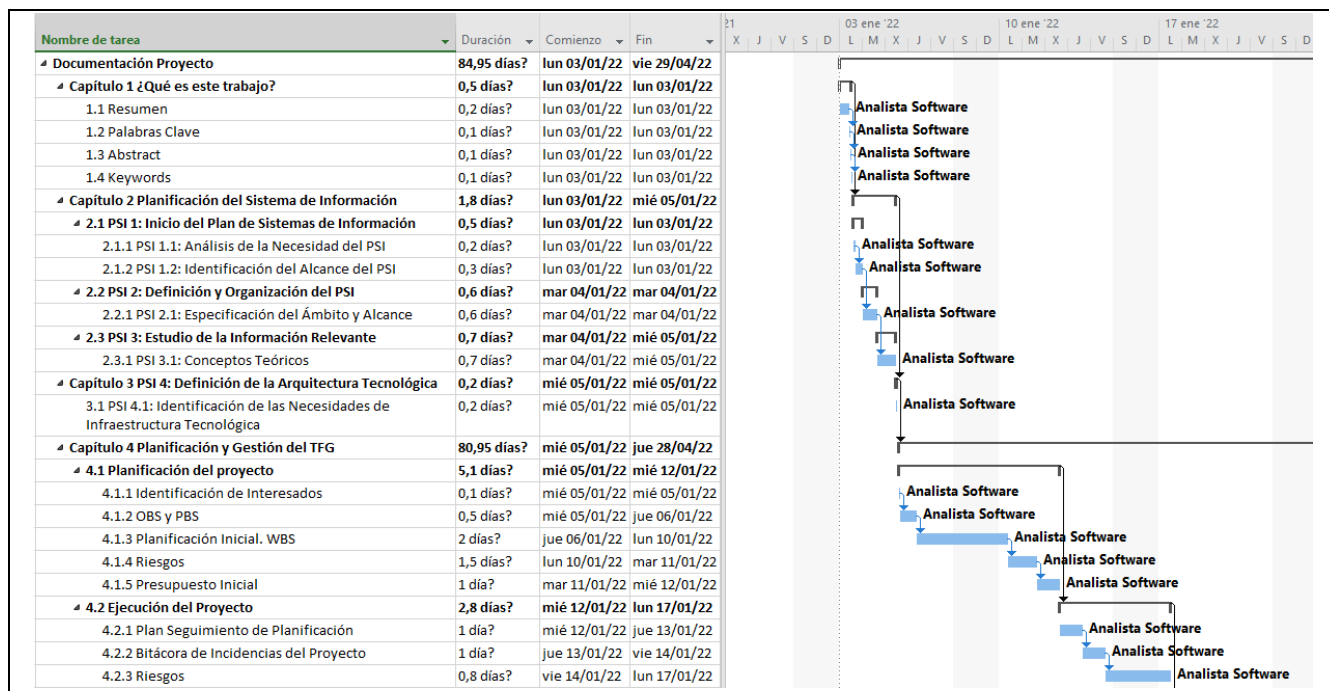
A fin de visualizar correctamente la *Work Breakdown Structure (WBS)* del proyecto, se mostrará primero una vista general, y después se verán los apartados en detalle en el orden de realización determinado. Además, cabe destacar que el archivo de MS Project que contiene la planificación figura en la carpeta de anexos del proyecto.



Vista general de la planificación del proyecto

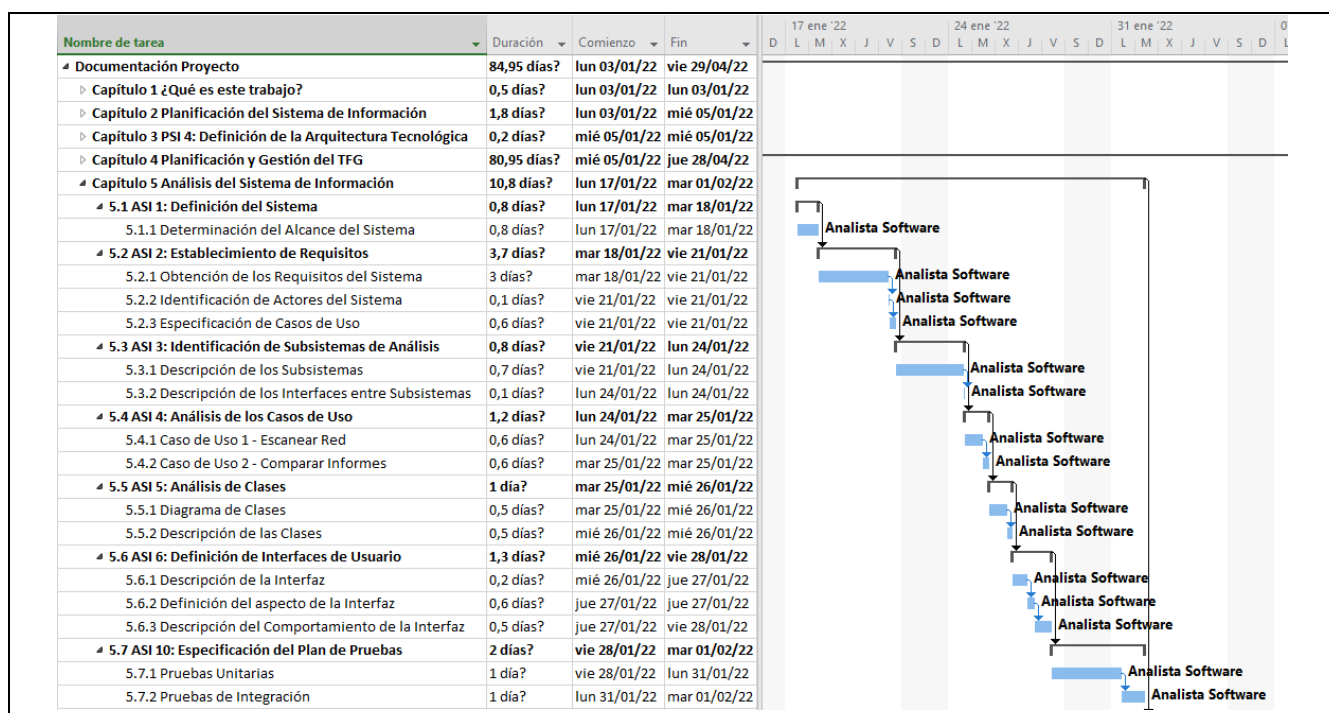
A la izquierda de la imagen anterior tenemos las tareas generales que conforman el proyecto, con su tiempo de duración estimado y las fechas en las que se realizarían; mientras que a la derecha hay un diagrama de Gantt que permite visualizar el orden y las relaciones entre ellas. Podemos ver que el proyecto empezaría el 3 de enero de 2022 y acabaría el 28 de abril de 2022, lo que supone casi 4 meses de trabajo.

Entre las tareas del proyecto se encuentran la redacción de la documentación, dividida en capítulos, el desarrollo de la aplicación, y el testeo.



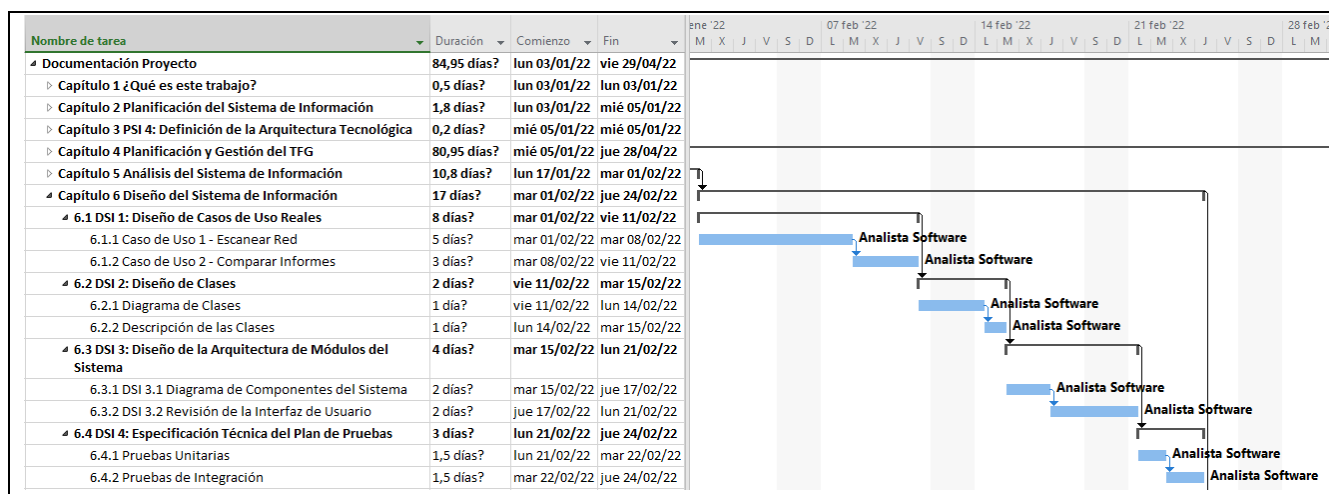
Planificación del proyecto para los capítulos 1-4 de la documentación

El proyecto empezaría por los primeros capítulos de la documentación, en los que se determinarían las líneas generales a seguir (funcionalidades requeridas a bajo nivel de detalle) y cómo se va a llevar a cabo (planificación de tareas y determinar los recursos necesarios). Esta parte la llevará a cabo el analista software y duraría unos 13 días.



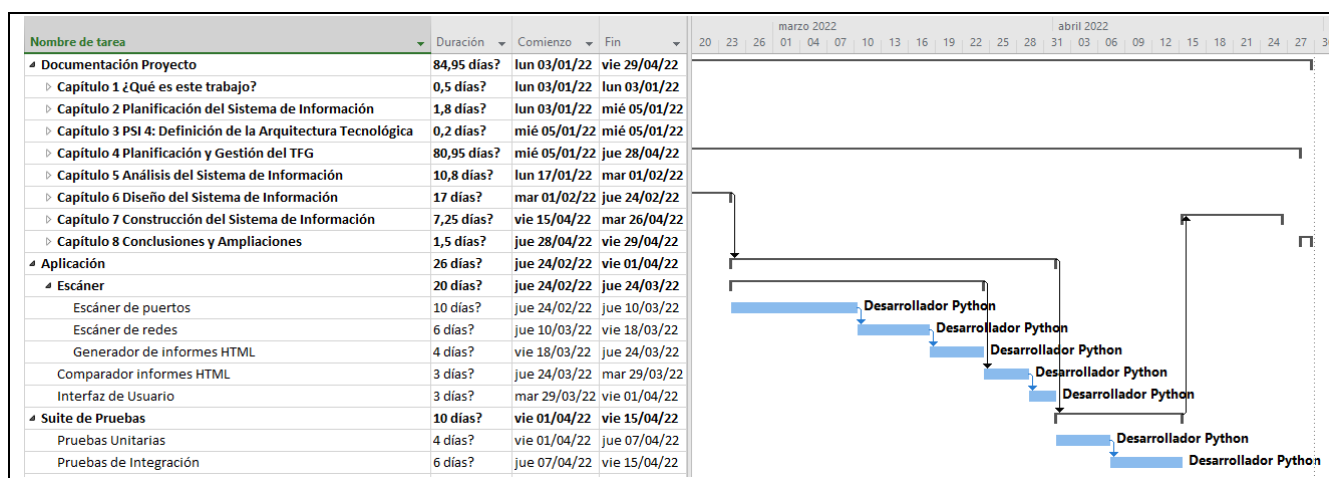
Planificación del proyecto para el capítulo 5 de la documentación

Tras tener claro lo básico del proyecto, empezaría la fase de análisis, en la que el analista software daría un nivel de detalle mayor a las ideas especificadas anteriormente (requisitos del sistema, casos de uso, clases, interfaces de usuario, pruebas...). Esta fase duraría unos 14 días.



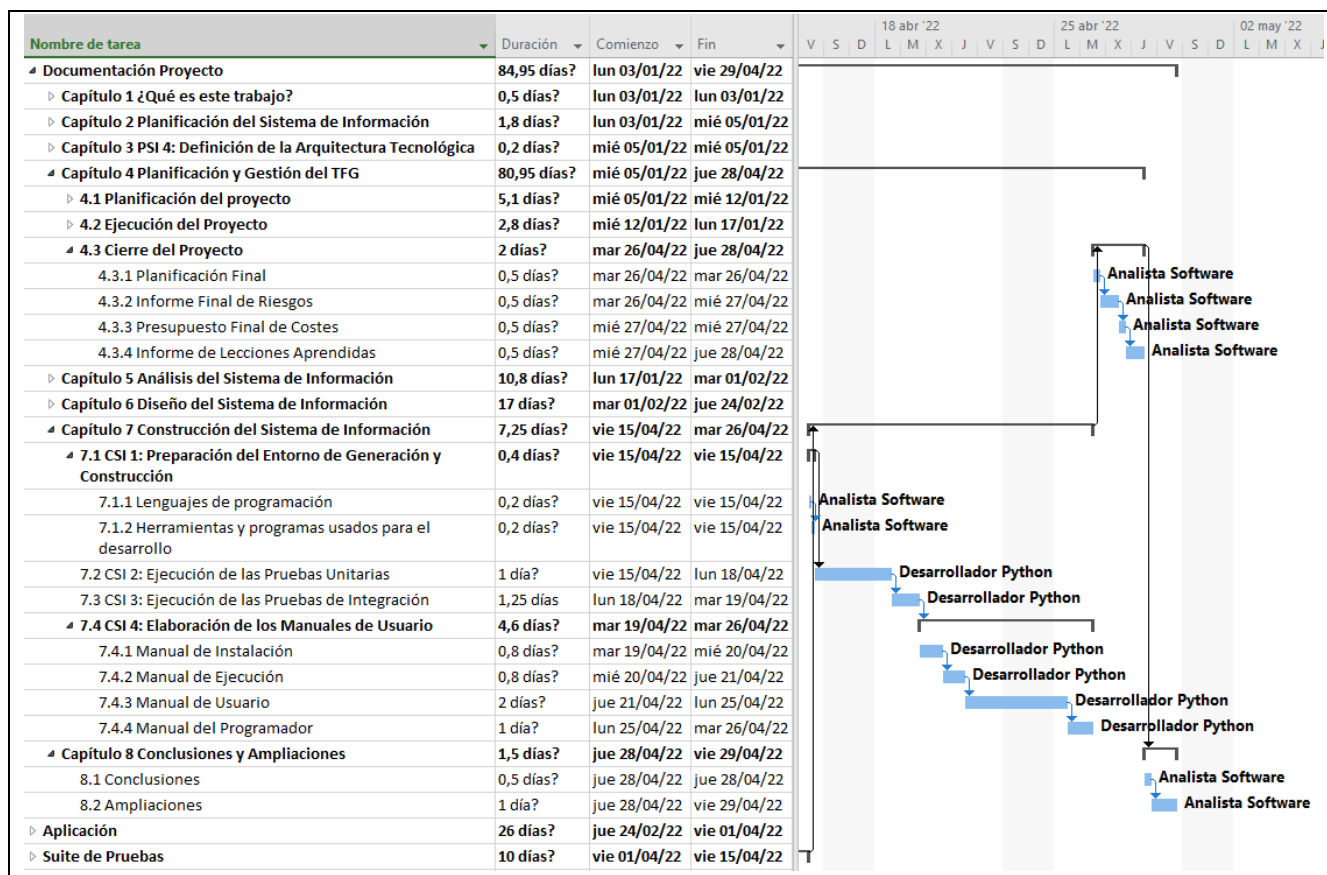
Planificación del proyecto para el capítulo 6 de la documentación

La fase de análisis da pie a la fase de diseño del sistema, en la que el analista software especificará todo lo necesario para obtener un diseño que pueda implementar el desarrollador Python (arquitectura del sistema, clases, plan de pruebas, interfaces...). Esta fase duraría unos 21 días.



Planificación del proyecto para el desarrollo de la aplicación y de las pruebas

Con el diseño de la aplicación listo, el desarrollador Python puede empezar a construir la aplicación en sí, empezando por los escáneres, que son la parte fundamental del sistema. Una vez completada, se desarrollaría la suite de pruebas. Estas dos partes llevarían en total unos 36 días.



Planificación del proyecto para el capítulo 7 de la documentación, el cierre del proyecto, y el capítulo 8

Después llegaríamos al capítulo 7 (constituido por la ejecución de las pruebas, los manuales de la aplicación...), que se desarrollaría principalmente por el desarrollador Python. Después el analista software volvería al capítulo 4, en el que dejamos el apartado 3 sin completar puesto que no tiene sentido hacerlo hasta llegar a este punto del proyecto. Finalmente, el analista software cerraría el proyecto acabando el capítulo 8.

4.1.4 Riesgos

4.1.4.1 Plan de Gestión de Riesgos

El contenido de este apartado se encuentra en el anexo “Plan de Gestión de Riesgos”.

4.1.4.2 Identificación de Riesgos

1. **La estimación de tiempo de las tareas fue demasiado optimista.** Una visión demasiado optimista de las capacidades de los perfiles del proyecto o de los esfuerzos requeridos para cumplir las tareas puede hacer que el tiempo asignado a las tareas sea menor del que debería.
2. **El desarrollador del proyecto se pone enfermo de COVID.** La pandemia de COVID que azota el mundo entero podría afectar al desarrollador del proyecto, haciendo que esté indisponible para hacer las tareas asignadas en el marco de tiempo especificado.
3. **El repositorio del proyecto es hackeado.** Si sucediera una brecha de seguridad el código de la aplicación podría verse comprometido, haciendo que se pudiera perder el trabajo desarrollado hasta la fecha o se tuviera que pagar un rescate.
4. **Entender mal un requisito del proyecto.** Si uno de los requisitos del proyecto se ha entendido mal y no se detecta en el momento, podría desencadenar una serie de cambios en cascada, haciendo que haya que rehacer varias partes del proyecto y teniendo que dedicar tiempo y esfuerzos extra para arreglarlas.
5. **La estación de trabajo del desarrollador deja de funcionar.** Si la máquina en la que trabaja el desarrollador deja de funcionar (no se enciende, va excesivamente lenta, se congela...), el desarrollador no puede cumplir sus tareas asignadas a tiempo y la planificación del proyecto podría verse comprometida. Esto podría suceder si el hardware usado es de poca calidad, tiene muchos años, o se le pide más rendimiento que para el que está preparado.
6. **Cambio en la especificación de una cabecera estándar existente.** Si el estándar de cabeceras cambiase y se empezaran a implementar cabeceras de una forma no esperada, la herramienta no podría clasificar estas correctamente.
7. **Aparición de nuevas cabeceras no consideradas hasta ahora.** Y al igual que en el caso anterior, las nuevas cabeceras no podrían ser clasificadas correctamente.
8. **Sistemas anti-escáner de una máquina objetivo bloquean las peticiones de la herramienta.** Es común prohibir las peticiones que no provengan de un navegador estándar, lo cual nos impediría analizar el servicio web.
9. **Un puerto HTTP ofrece un servicio no HTTP.** Analizar este puerto podría causar problemas en el servicio al no estar preparado para recibir peticiones HTTP, y además la herramienta podría recibir respuestas para las que no está preparada.
10. **Latencias de red elevadas.** Una latencia de red elevada puede provocar que máquinas objetivo que estén disponibles no sean analizadas debido a timeouts, haciendo un análisis incompleto.

4.1.4.3 Registro de Riesgos

ID	Nombre del Riesgo	Categoría	Probabilidad	Impacto				Prioridad
				Coste	Tiempo	Alcance	Calidad	
1	La estimación del tiempo de las tareas fue demasiado optimista	Técnico	Alta	Alto	Alto	Bajo	Bajo	0,39
2	El desarrollador del proyecto se pone enfermo de COVID	Organizacional	Baja	Alto	Crítico	Alto	Medio	0,27
3	El repositorio del proyecto es hackeado	Gestión del Proyecto	Muy Baja	Crítico	Crítico	Crítico	Crítico	0,09
4	Entender mal un requisito del proyecto	Técnico	Media	Bajo	Medio	Medio	Medio	0,15
5	La estación de trabajo del desarrollador deja de funcionar	Organizacional	Baja	Medio	Alto	Alto	Alto	0,17
6	Cambio en la especificación de una cabecera estándar existente	Externo	Muy Baja	Inapreciable	Inapreciable	Medio	Inapreciable	0,03
7	Aparición de nuevas cabeceras no consideradas hasta ahora	Externo	Media	Inapreciable	Bajo	Bajo	Bajo	0,08
8	Sistemas anti-escáner de una máquina objetivo bloquean las peticiones de la herramienta	Técnico	Media	Bajo	Medio	Medio	Alto	0,28
9	Un puerto HTTP ofrece un servicio no HTTP	Técnico	Muy Baja	Medio	Medio	Bajo	Medio	0,03
10	Latencias de red elevadas	Técnico	Baja	Bajo	Medio	Bajo	Alto	0,17

Desglose del cálculo de la prioridad de los distintos riesgos del proyecto

ID	Nombre del Riesgo	Respuesta al Riesgo	Estrategia
1	La estimación del tiempo de las tareas fue demasiado optimista	Asignar tiempo extra a las tareas críticas del proyecto para que al menos lo importante se gestione con el tiempo adecuado	Mitigar el riesgo
2	El desarrollador del proyecto se pone enfermo de COVID	No se puede controlar al 100% la exposición del desarrollador al virus, por lo que solo podemos aceptar que es posible que suceda	Asumir el riesgo
3	El repositorio del proyecto es hackeado	Aunque ningún sistema informático es completamente seguro, podemos reducir la probabilidad de que un tercero acceda al repositorio cambiando las contraseñas de la máquina de desarrollo y del repositorio una vez al mes	Mitigar el riesgo
4	Entender mal un requisito del proyecto	Asegurarse de que lo que quiere el cliente es lo que se está haciendo mediante múltiples reuniones de control y un seguimiento exhaustivo	Eliminar el riesgo
5	La estación de trabajo del desarrollador deja de funcionar	Hacer una revisión de la máquina semanalmente y encargar recambios preventivamente para poder hacer los cambios necesarios con el mínimo tiempo	Mitigar el riesgo
6	Cambio en la especificación de una cabecera estándar existente	Si cambiase la especificación solo tendríamos que actualizar el diccionario de cabeceras, así que no hace falta prepararse para ello	Asumir el riesgo
7	Aparición de nuevas cabeceras no consideradas hasta ahora	Si apareciesen cabeceras nuevas solo tendríamos que añadirlas al diccionario de cabeceras, así que no hace falta prepararse para ello	Asumir el riesgo
8	Sistemas anti-escáner de una máquina objetivo bloquean las peticiones de la herramienta	Configurar las peticiones HTTP que mande la herramienta para que pueda hacerse pasar por un navegador web estándar mediante la cabecera 'User-Agent'	Eliminar el riesgo
9	Un puerto HTTP ofrece un servicio no HTTP	Es responsabilidad del dueño del servicio web prepararlo para peticiones inesperadas	Transferir el riesgo
10	Latencias de red elevadas	Preparar la herramienta para estos casos poco frecuentes la haría mucho más lenta, por lo que no merece la pena	Asumir el riesgo

Estrategias a aplicar para minimizar los riesgos del proyecto

4.1.5 Presupuesto Inicial

4.1.5.1 Presupuesto de Costes

Como en el proyecto no hay que comprar materiales o contratar servicios externos, los costes se resumen en el número de horas asignadas a cada perfil de desarrollo multiplicadas por su precio/hora correspondiente. El desglose de horas asignadas se puede ver en la estructura de desglose de tareas (WBS), mientras que aquí solo figura el total. Los precios/hora se han obtenido a partir de la media de los sueldos de 10 ofertas de trabajo en InfoJobs de perfiles similares a los que requiere el proyecto.

Personal	Horas Asignadas	Precio/Hora	Coste Total
Analista Software	336,8 horas	25€/Hora	8.420€
Desarrollador Python	342,8 horas	18€/Hora	6.170,40€
			14.590,40€

Tras los cálculos previos, el presupuesto inicial del proyecto asciende a 14.590,40€.

4.2 EJECUCIÓN DEL PROYECTO

4.2.1 Plan de Seguimiento de la Planificación

Se han establecido tres líneas base para hacer un seguimiento de la planificación del proyecto:

4.2.1.1 Línea base inicial

La línea base inicial se estableció al principio al comienzo del proyecto, a fecha de 3 de enero.

4.2.1.2 Línea base a medio proyecto

La segunda línea base se estableció después de la construcción de la parte de la documentación necesaria para desarrollar la aplicación (con “6.4 DSI 4: Especificación Técnica del Plan de Pruebas” como última tarea), el 24 de febrero. En el apartado “Planificación Final” se pueden ver en detalle la comparación de los tiempos planificados y los reales.

4.2.1.3 Línea base de final de proyecto

Finalmente, la tercera línea base se estableció tras completar la aplicación y la documentación, más concretamente, una vez acabada la última tarea del proyecto (“8.2 Ampliaciones”), el 11 de mayo. Al igual que con la anterior línea base, los detalles de los tiempos están disponibles en el apartado “Planificación Final”.

4.2.2 Bitácora de Incidencias del Proyecto

- **2/3/2022:** Durante el desarrollo del analizador de servicios web se detectó un comportamiento inconsistente al hacer análisis de una máquina de prueba. El problema, surgido por la falta de algunos conocimientos de sockets a bajo nivel, provocó que se tuviera que dedicar tiempo extra (dos días) a estudiar documentación específica y a reestructurar partes del código para que funcionase todo como se esperaba.
- **9/3/2022:** El analizador de servicios web empezó a funcionar mal al analizar servidores HTTPS debido a un problema con el certificado SSL usado por la aplicación. Para arreglarlo hubo que usar una librería nueva y dedicar dos días extra a re-implementar lo que fallaba.
- **6/4/2022:** El framework usado para desarrollar la interfaz de usuario demostró ser más complicado de lo que parecía en un principio. Esto, añadido a sus particularidades, provocó una serie de problemas durante la implementación que tuvieron que ser arreglados tras estudiar la documentación correspondiente, consumiendo un día más de lo planeado.

- **19/4/2022:** A la hora de implementar las pruebas de integración del sistema, nos encontramos con que no hay una solución idónea para testear interfaces de usuario en Python (como podría ser Selenium para interfaces web) y que la mejor forma de hacerlo pasa por llevar a cabo un proceso complicado y propenso a errores. Por estos motivos hizo falta un día extra de trabajo.

4.2.3 Riesgos

Se ha hecho el seguimiento de los 5 riesgos con más prioridad:

ID	Nombre del riesgo	¿Ocurrió?	Comentarios
1	La estimación de tiempo de las tareas fue demasiado optimista	Sí	Aunque se mitigó el riesgo añadiendo tiempo extra a las tareas críticas, hubo otras tareas menores que necesitaron más tiempo para poder ser completadas
8	Sistemas anti-escáner de una máquina objetivo bloquean las peticiones de la herramienta	No	Tras identificar este riesgo y preparar la aplicación para ello, fue eliminado satisfactoriamente
2	El desarrollador del proyecto se pone enfermo de COVID	No	Hubo dos ocasiones en las que el desarrollador pudo haberse puesto enfermo, pero afortunadamente no sucedió
5	La estación de trabajo del desarrollador deja de funcionar	No	El mantenimiento semanal de la estación de desarrollo ha procurado que siempre haya estado disponible
10	Latencias de red elevadas	No	Como hemos asumido este riesgo no se ha preparado la aplicación para ello, sin embargo durante las pruebas realizadas no ha surgido

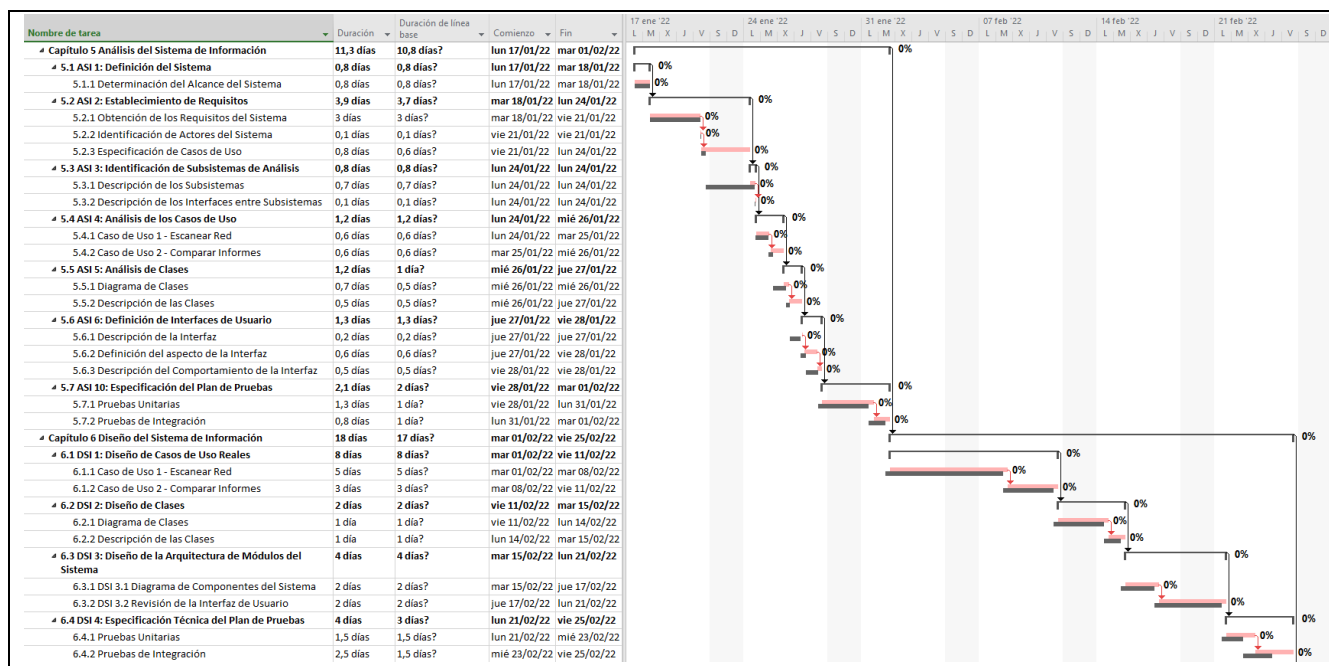
4.3 CIERRE DEL PROYECTO

4.3.1 Planificación Final

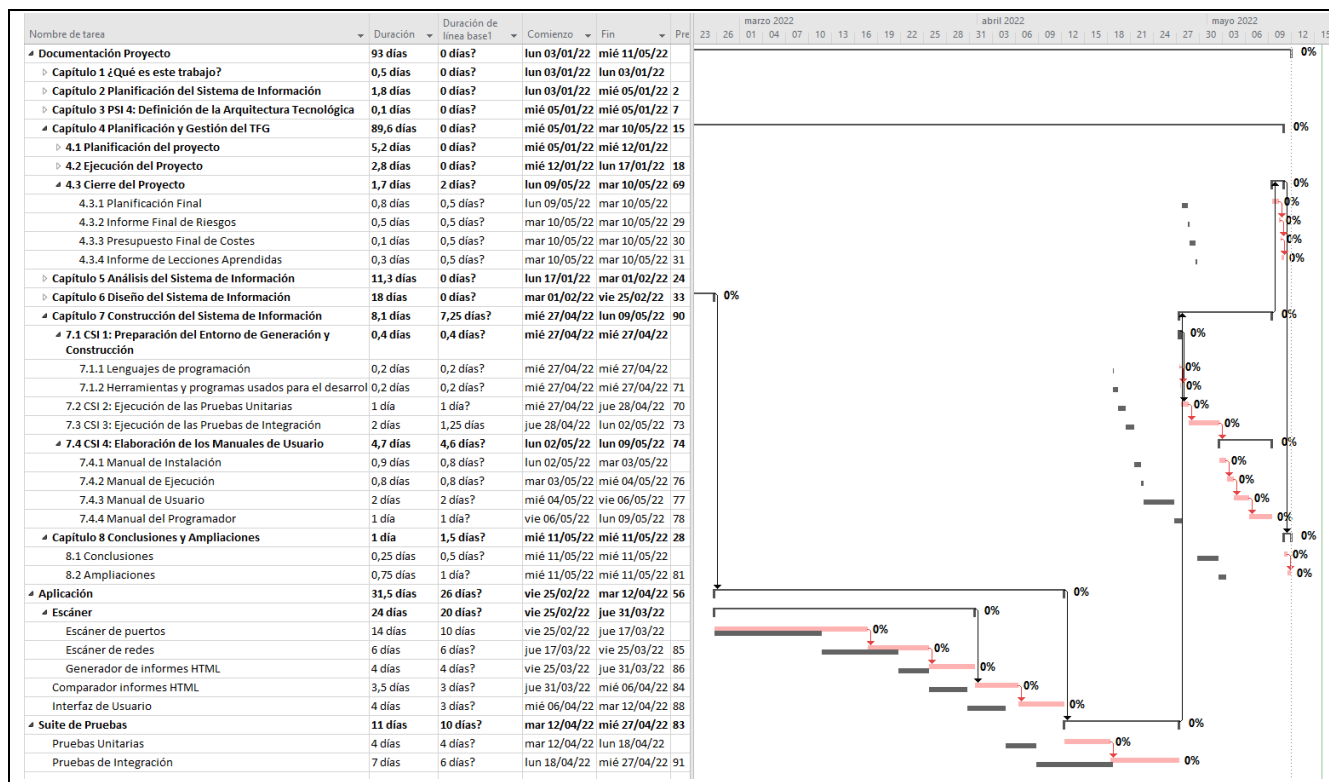
A continuación se puede ver la planificación final del proyecto, con la duración y fecha reales de las tareas respecto a su duración y fecha planificadas, a la derecha gráficamente (gris para el tiempo planificado, rojo para el real), y a la izquierda numéricamente (Columna 'Duración' para la real y 'Duración de línea base' para la planificada):

Nombre de tarea	Duración	Duración de línea base	Comienzo	Fin	J	V	S	D	03 ene '22	10 ene '22	17 ene '22
Documentación Proyecto	86,75 días?	84,95 días?	lun 03/01/22	mar 03/05/22							
Capítulo 1 ¿Qué es este trabajo?	0,5 días	0,5 días?	lun 03/01/22	lun 03/01/22							
1.1 Resumen	0,2 días	0,2 días?	lun 03/01/22	lun 03/01/22							
1.2 Palabras Clave	0,1 días	0,1 días?	lun 03/01/22	lun 03/01/22							
1.3 Abstract	0,1 días	0,1 días?	lun 03/01/22	lun 03/01/22							
1.4 Keywords	0,1 días	0,1 días?	lun 03/01/22	lun 03/01/22							
Capítulo 2 Planificación del Sistema de Información	1,8 días	1,8 días?	lun 03/01/22	mié 05/01/22							
2.1 PSI 1: Inicio del Plan de Sistemas de Información	0,4 días	0,5 días?	lun 03/01/22	lun 03/01/22							
2.1.1 PSI 1.1: Análisis de la Necesidad del PSI	0,2 días	0,2 días?	lun 03/01/22	lun 03/01/22							
2.1.2 PSI 1.2: Identificación del Alcance del PSI	0,2 días	0,3 días?	lun 03/01/22	lun 03/01/22							
2.2 PSI 2: Definición y Organización del PSI	0,8 días	0,6 días?	lun 03/01/22	mar 04/01/22							
2.2.1 PSI 2.1: Especificación del Ámbito y Alcance	0,8 días	0,6 días?	lun 03/01/22	mar 04/01/22							
2.3 PSI 3: Estudio de la Información Relevante	0,6 días	0,7 días?	mar 04/01/22	mié 05/01/22							
2.3.1 PSI 3.1: Conceptos Teóricos	0,6 días	0,7 días?	mar 04/01/22	mié 05/01/22							
Capítulo 3 PSI 4: Definición de la Arquitectura Tecnológica	0,1 días	0,2 días?	mié 05/01/22	mié 05/01/22							
3.1 PSI 4.1: Identificación de las Necesidades de Infraestructura	0,1 días	0,2 días?	mié 05/01/22	mié 05/01/22							
Capítulo 4 Planificación y Gestión del TFG	82,85 días?	80,95 días?	mié 05/01/22	lun 02/05/22							
4.1 Planificación del proyecto	5,2 días	5,1 días?	mié 05/01/22	mié 12/01/22							
4.1.1 Identificación de Interesados	0,1 días	0,1 días?	mié 05/01/22	mié 05/01/22							
4.1.2 OBS y PBS	0,8 días	0,5 días?	mié 05/01/22	jue 06/01/22							
4.1.3 Planificación Inicial. WBS	2 días	2 días?	jue 06/01/22	lun 10/01/22							
4.1.4 Riesgos	1,7 días	1,5 días?	lun 10/01/22	mar 11/01/22							
4.1.5 Presupuesto Inicial	0,6 días	1 día?	mié 12/01/22	mié 12/01/22							
4.2 Ejecución del Proyecto	2,8 días	2,8 días?	mié 12/01/22	lun 17/01/22							
4.2.1 Plan Seguimiento de Planificación	1 día	1 día?	mié 12/01/22	jue 13/01/22							
4.2.2 Bitácora de Incidencias del Proyecto	1 día	1 día?	jue 13/01/22	vie 14/01/22							
4.2.3 Riesgos	0,8 días	0,8 días?	vie 14/01/22	lun 17/01/22							

***Duración y fecha reales vs planificadas de las tareas de los capítulos 1-4 de la documentación
(línea base a medio proyecto)***



**Duración y fecha reales vs planificadas de las tareas de los capítulos 5-6 de la documentación
(línea base a medio proyecto)**



Duración y fecha reales vs planificadas de las tareas de construcción de la aplicación y los capítulos restantes de la documentación

Si bien es cierto que para la gran mayoría de tareas se ha cumplido su planificación, ha habido una serie de ellas para las que ha habido que ajustar ligeramente los tiempos. Afortunadamente, las tareas que requirieron menos tiempo equilibraron más o menos las que requirieron tiempo extra. Sin embargo, hubo otras tareas durante la fase de desarrollo de la aplicación y de la fase de testeo que requirieron bastante tiempo adicional, retrasando de forma significativa el final del proyecto.

El proyecto empezó el 3 de enero de 2022 y terminó el 11 de mayo de 2022, lo que supone una duración total de aproximadamente 4 meses y medio de trabajo.

4.3.2 Informe Final de Riesgos

No se han identificado nuevos riesgos o modificaciones de los riesgos identificados al principio del proyecto o durante su ejecución, así que cualquier detalle sobre ellos puede encontrarse en los apartados 4.1.4 y 4.2.3.

4.3.3 Presupuesto Final de Costes

Al igual que en el presupuesto inicial (apartado 4.1.5), el coste del proyecto viene determinado por las horas de trabajo que requiere cada uno de los perfiles del proyecto. Como mencionamos antes, el proyecto ha requerido más horas de las esperadas, lo que se refleja en el coste total del proyecto:

Personal	Horas Asignadas	Precio/Hora	Coste Total
Analista Software	342,4 horas	25€/Hora	8.560€
Desarrollador Python	401,6 horas	18€/Hora	7.228,80€
			15.788,80€

Tras los cálculos previos, el presupuesto final del proyecto pasa de 14.590,40€ a 15.788,80€.

4.3.4 Informe de Lecciones Aprendidas

- El conocimiento de las tecnologías a usar es un factor clave a la hora de desarrollar software, y merece la pena dedicar tiempo a aprender todo lo que va a ser relevante antes de empezar a trabajar para no malgastar tiempo más adelante.
- La planificación es clave para que el proyecto se desarrolle adecuadamente, pero por muy claro que uno tenga los conceptos teóricos de ella, es muy difícil hacerlo todo lo mejor posible sin mucha, mucha experiencia práctica planificando.
- Los imprevistos y problemas van a surgir durante un proyecto y, aunque es importante dedicarles tiempo y esforzarse, a veces la mejor opción es ver qué se puede hacer al respecto con la cabeza despejada.

Capítulo 5. ANÁLISIS DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

ASI

5.1 ASI 1: DEFINICIÓN DEL SISTEMA

5.1.1 Determinación del Alcance del Sistema

El objetivo del proyecto es desarrollar una aplicación de escritorio que pueda escanear una red de máquinas para encontrar los hosts activos, luego ubique todos los servicios web que ofrece cada uno de ellos y después analice qué cabeceras HTTP se encuentran implementadas o no en cada uno de ellos. Estos tres tipos de escáner no podrán ser accedidos de forma individual, de forma que no se podrá escanear la red solo para encontrar los hosts activos, escanear una máquina sin determinar primero si está activa (mediante el escáner de redes) o analizar los valores de cabeceras HTTP a partir de un archivo de texto o similar. El usuario solo podrá determinar un rango de redes a escanear, que será facilitado al escáner de redes, éste a su vez facilitará la lista de hosts activos al escáner de máquinas, y éste a su vez facilitará la lista de puertos HTTP/HTTPS al escáner de puertos.

La información obtenida durante el proceso de escaneo tiene que sintetizarse en uno o más informes que incluyan todos los detalles del análisis de forma comprensible junto con recomendaciones para resolver los problemas que haya encontrado. Estos informes solo se generarán en un único formato (HTML), y el usuario no podrá determinar los colores del archivo, el idioma del texto, o qué número de informes se generan (siempre uno por máquina y uno que incluya la información de todos en conjunto).

Además, estos informes tienen que poder compararse entre sí, de forma que se pueda ver la evolución de un servicio web entre dos análisis en momentos distintos en un informe nuevo. En este informe tampoco se podrán personalizar los colores del archivo ni el idioma del texto.

Finalmente, estas funcionalidades tienen que ser accesibles desde una interfaz de usuario, que solo estará disponible en un idioma por defecto (inglés).

5.2 ASI 2: ESTABLECIMIENTO DE REQUISITOS

5.2.1 Obtención de los Requisitos del Sistema

5.2.1.1 Escaneo de redes

RF_ER1 El sistema deberá poder determinar de forma automática qué máquinas se encuentran activas en un rango de red determinado:

RF_ER1.1 El rango de red se determinará siguiendo el formato CIDR

5.2.1.2 Escaneo de servicios web

RF_ESW1 El sistema deberá poder analizar de forma automática los puertos de una máquina activa para encontrar los servicios web que ofrece.

RF_ESW1.1 El sistema deberá poder escanear una máquina si comparten una misma red.

RF_ESW1.2 El sistema deberá poder determinar qué puertos de una máquina hospedan servicios web.

RF_ESW1.3 El sistema deberá obtener las cabeceras HTTP que implementan cada uno de los servicios web de la máquina escaneada.

RF_ESW1.3.1 El sistema deberá guardar esta información de forma que se pueda sintetizar en un informe más tarde.

RF_ESW1.3.2 El sistema deberá establecer qué cabeceras son relevantes para la seguridad del servicio y cuáles no.

5.2.1.3 Generación de informes de escaneos

RF_GIE1 El sistema deberá sintetizar la información obtenida durante los escaneos en informes generados de forma automática.

RF_GIE1.1 El sistema deberá generar informes en formato HTML.

RF_GIE1.2 El sistema deberá señalar qué cabeceras ha encontrado en cada servicio web.

RF_GIE1.3 El sistema deberá incluir referencias externas de cada uno de los tipos de cabeceras que encuentre.

RF_GIE1.4 El sistema deberá señalar qué problemas ha encontrado en cada servicio web:

RF_GIE1.4.1 El sistema deberá señalar qué cabeceras que no estén implementadas en un servicio web deberían estarlo. Esto incluye:

RF_GIE1.4.1.1 Las cabeceras que representen medidas básicas de seguridad

RF_GIE1.4.2 El sistema deberá señalar qué cabeceras que estén implementadas en un servicio web no deberían estarlo. Esto incluye:

RF_GIE1.4.2.1 Las cabeceras que estén deprecadas.

RF_GIE1.4.2.2 Las cabeceras que estén en fase experimental.

RF_GIE1.4.3 El sistema deberá señalar qué cabeceras que estén implementadas en un servicio web están mal implementadas. Esto incluye:

RF_GIE1.4.3.1 Las cabeceras que incluyan valores que pueden suponer un peligro de seguridad.

RF_GIE2 El sistema deberá generar un número determinado de informes dependiendo de la cantidad de máquinas encontradas con servicios web:

RF_GIE2.1 Si encuentra una sola máquina, el sistema deberá generar un único informe.

RF_GIE2.2 Si encuentra dos o más máquinas, el sistema deberá generar un informe individual de cada una de ellas y además uno en el que estén todos los resultados juntos.

5.2.1.4 Comparación de informes

RF_CI1 El sistema deberá permitir comparar informes de servicios web generados previamente por el propio sistema.

RF_CI1.1 El sistema deberá generar un informe de comparación a partir de dos informes a comparar, de forma que se diferencien entre sí:

RF_CI1.1.1 Los valores que se hayan añadido.

RF_CI1.1.2 Los valores que se hayan eliminado.

RF_CI1.1.3 Los valores que se hayan modificado.

5.2.1.5 Interfaz de Usuario

RF_IU1 El sistema deberá incluir una interfaz de usuario que permita acceder a las funcionalidades de la herramienta:

RF_IU1.1 La interfaz de usuario deberá incluir una ventana para acceder al escáner de redes, que deberá incluir:

RF_IU1.1.1 Un campo de texto que explique cómo usar el escáner de redes.

RF_IU1.1.2 Un campo que acepte un rango de direcciones IP en formato CIDR.

RF_IU1.1.3 Un botón para iniciar el escaneo de máquinas en el rango de IPs proporcionado por el usuario.

RF_IU1.2 La interfaz de usuario deberá incluir una ventana para acceder al comparador de informes, que deberá incluir:

RF_IU1.2.1 Un campo de texto que explique cómo usar el comparador de informes.

RF_IU1.2.2 Un botón que abra un explorador de archivos para seleccionar el primer informe.

RF_IU1.2.3 Un botón que abra un explorador de archivos para seleccionar el segundo informe.

RF_IU1.2.4 Un botón que ejecute el comparador de informes sobre los dos informes seleccionados.

RF_IU2 La interfaz de usuario deberá incluir una ventana de texto en el que se notifique:

RF_IU2.1 El progreso de las operaciones que estén en curso.

RF_IU2.2 Los problemas que encuentre en la ejecución de las operaciones.

5.2.1.6 Requisitos no funcionales

RNF_USR1 El usuario deberá tener los siguientes conocimientos para manejar la aplicación:

RNF_USR1.1 Funcionamiento del protocolo IP.

RNF_USR1.2 Bloques CIDR.

RNF_USR1.3 Funcionamiento del protocolo HTTP/HTTPS y los servicios web.

RNF_TCN1 El sistema deberá funcionar en una máquina que:

RNF_TCN1.1 Tenga Python 3.9 o superior.

RNF_TCN1.2 Tenga interfaz gráfica.

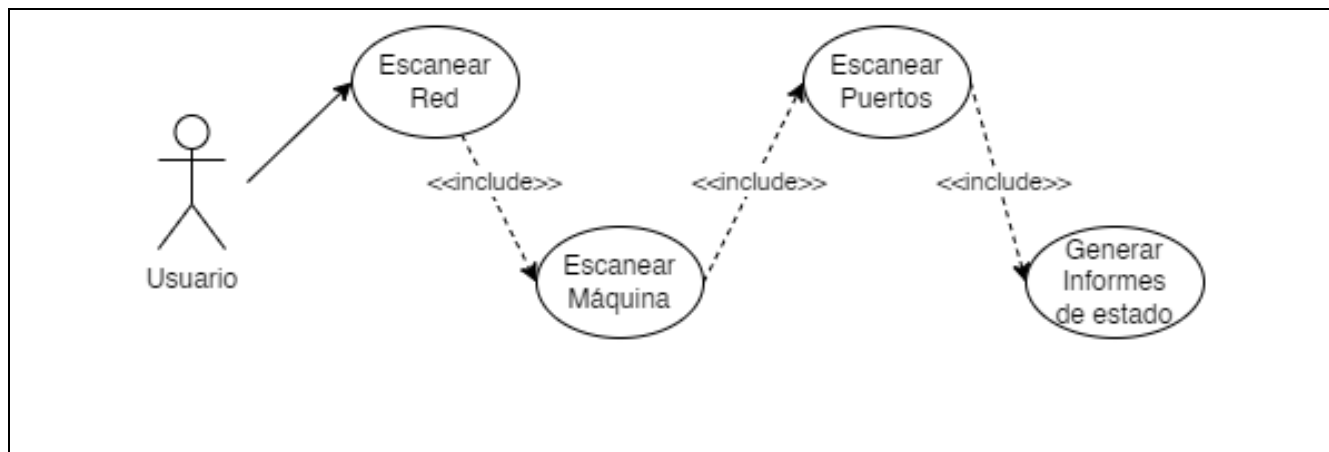
RNF_USBL1 Los usuarios del sistema deberán poder aprender a usarlo en 1 hora o menos.

5.2.2 Identificación de Actores del Sistema

En este sistema solo existe un actor, el usuario de la aplicación.

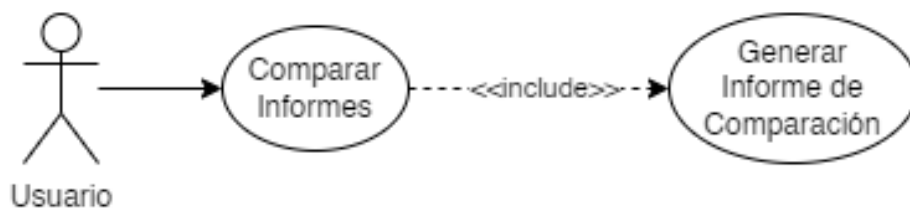
5.2.3 Especificación de Casos de Uso

En este apartado se muestran los casos de uso de la herramienta. Lo habitual es que haya varios de ellos, pero en este caso solo hay dos. El motivo es que la operación principal conlleva una serie de etapas encadenadas e inseparables, ya que se tienen que proporcionar los resultados de una a la siguiente sucesivamente.



Caso de uso 1 – Escanear Red

Nombre del Caso de Uso	
Escanear Red	
Descripción	
El programa mostrará al usuario en una ventana las instrucciones para escanear una red, que consisten en rellenar un campo de texto con un rango de direcciones IP en formato CIDR y pulsar un botón para iniciar el escaneo. Cuando complete ese proceso, si el texto introducido es incorrecto se mostrará un mensaje de error; si no, la herramienta escaneará el rango de red introducido, las máquinas que encuentre, y sus puertos para hacer los informes necesarios. Una vez generados los informes, se abrirán en el navegador por defecto de la máquina.	



Caso de uso 2 – Comparar Informes

Nombre del Caso de Uso
Comparar Informes
Descripción
El programa mostrará al usuario en una ventana las instrucciones para comparar dos informes, que consisten en seleccionar dos archivos HTML en un explorador de archivos y pulsar un botón para iniciar la comparación. Si tras dar al botón los dos archivos seleccionados son correctos, se realizará la comparación y se generará un nuevo informe, que se abrirá en el navegador por defecto de la máquina. Si no fuera así, se mostrará un mensaje de error con los detalles del problema encontrado.

5.3 ASI 3: IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS

5.3.1 Descripción de los Subsistemas

5.3.1.1 Interfaz de usuario

Este módulo desarrolla la forma que tiene el usuario de comunicarse con el sistema. Consiste en una interfaz gráfica que permite acceder al escáner de red o al comparador de informes, y un cuadro de texto que informa al usuario sobre el progreso de los procesos que realiza la herramienta. La parte del escáner de red incluye una forma de introducir el rango de redes IP a escanear, mientras que la del comparador, una manera de seleccionar dos archivos a comparar.

5.3.1.2 Escáner de red

Este módulo se encarga de analizar un rango de red para buscar qué hosts están activos en él, para luego buscar servicios web en cada uno de ellos. Por cada uno de los posibles puertos de las máquinas activas, analiza de qué tipo es y si sigue el protocolo HTTP/HTTPS envía una serie de peticiones. Si el puerto hospeda un servicio web, éste mandará una respuesta que incluirá las cabeceras que implementa el servicio. Esta información se procesa y se coteja con el diccionario de cabeceras HTTP, y con ello se genera un informe del análisis que incluye las cabeceras implementadas o las no implementadas y que deberían estarlo, con los valores de cada una de ellas, categorizadas, con referencias externas para entender más sobre ellas, y con partes resaltadas según contengan los valores esperados o no.

5.3.1.3 Comparador de informes

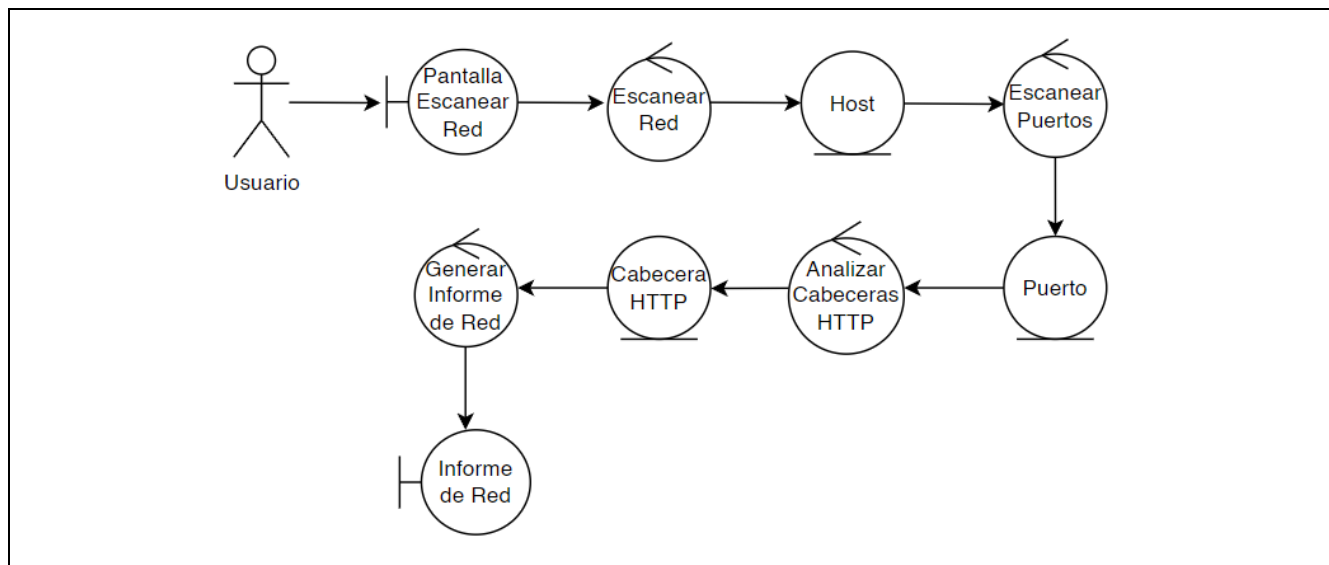
Este módulo recibe dos informes generados previamente con el escáner de red, los revisa, y marca en el segundo de ellos filas que se hayan añadido, eliminado, o modificado con respecto al primero, generando un informe nuevo en el que se pueden observar las diferencias entre ambos.

5.3.2 Descripción de los Interfaces entre Subsistemas

Solo existen dos interfaces entre los subsistemas: una entre la interfaz de usuario y el escáner de red, y otra entre la interfaz de usuario y el comparador de informes. Ambas suponen una comunicación local, en la que se transmite el texto introducido por el usuario: el rango de direcciones IP a escanear en la primera interfaz, y las direcciones en el sistema de archivos de los dos informes a comparar, en la segunda.

5.4 ASI 4: ANÁLISIS DE LOS CASOS DE USO

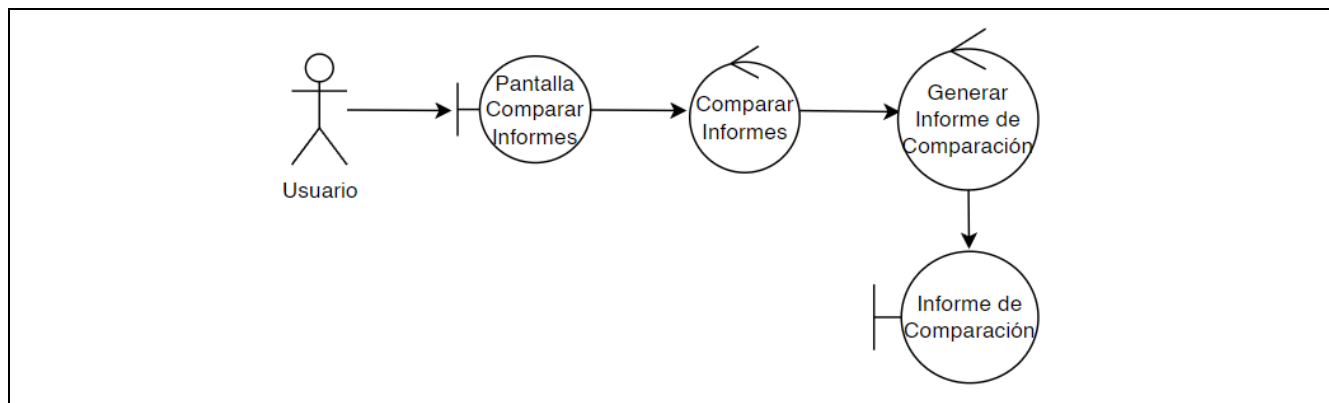
5.4.1 Caso de Uso 1 – Escanear Red



Descripción del “Caso de Uso 1 – Escanear Red” con un diagrama de robustez (I)

ESCANEAR RED	
PRECONDICIONES	El sistema no está escaneando una red o comparando informes
POSCONDICIONES	Si existe al menos un host con servicios web en la red escaneada, debe existir al menos un nuevo informe de red en el sistema de archivos de la máquina.
ACTORES	Iniciado y terminado por el usuario
DESCRIPCIÓN	<p>El usuario:</p> <ol style="list-style-type: none"> 1-Accederá a la pantalla de escanear red 2-Rellenará el rango de IPs a escanear 3-Lanzará el escáner con el rango especificado 4-Recibirá actualizaciones del progreso de las distintas operaciones involucradas 5-Verá cómo los informes generados se abren en su navegador
VARIACIONES (ESCENARIOS SECUNDARIOS)	<p>Escenario Alternativo 1: El rango de IPs a escanear no sigue un formato correcto</p> <p>Volver al paso 1 del escenario principal, notificar los problemas encontrados en la información introducida</p>
EXCEPCIONES	-

5.4.2 Caso de Uso 2 – Comparar Informes



Descripción del “Caso de Uso 2 – Comparar Informes” con un diagrama de robustez (II)

COMPARAR INFORMES	
PRECONDICIONES	El sistema no está escaneando una red o comparando informes
POSCONDICIONES	Debe existir un nuevo informe de comparación en el sistema de archivos de la máquina
ACTORES	Iniciado y finalizado por el usuario
DESCRIPCIÓN	<p>El usuario:</p> <ol style="list-style-type: none"> 1-Accederá a la pantalla de comparar informes 2-Seleccionará los informes a comparar mediante una ventana de exploración de archivos 3-Lanzará el comparador con los archivos seleccionados 4-Recibirá actualizaciones del progreso de las distintas operaciones involucradas 5-Verá cómo el informe generado se abre en su navegador
VARIACIONES (ESCENARIOS SECUNDARIOS)	<p>Escenario Alternativo 1: La dirección de uno o de los dos archivos es inválida</p> <p>Notificar qué problemas ha encontrado en las direcciones de los archivos y volver al paso 1 del escenario principal</p>
EXCEPCIONES	-

5.5 ASI 5: ANÁLISIS DE CLASES

5.5.1 Diagrama de Clases

A continuación se muestra el diagrama de clases del proyecto que, como se puede apreciar, es muy sencillo. Esto se debe a que la complejidad real reside en la funcionalidad del código y no en el modelo de datos de las entidades a manejar, que son solo tres. La responsabilidad de las clases es solo representar la información encontrada para su posterior tratamiento de forma cómoda. Técnicamente sería posible conseguir la misma funcionalidad sin clases en absoluto, pero de esa forma nos perderíamos una serie de ventajas que aporta la programación orientada a objetos, dificultando significativamente el desarrollo.

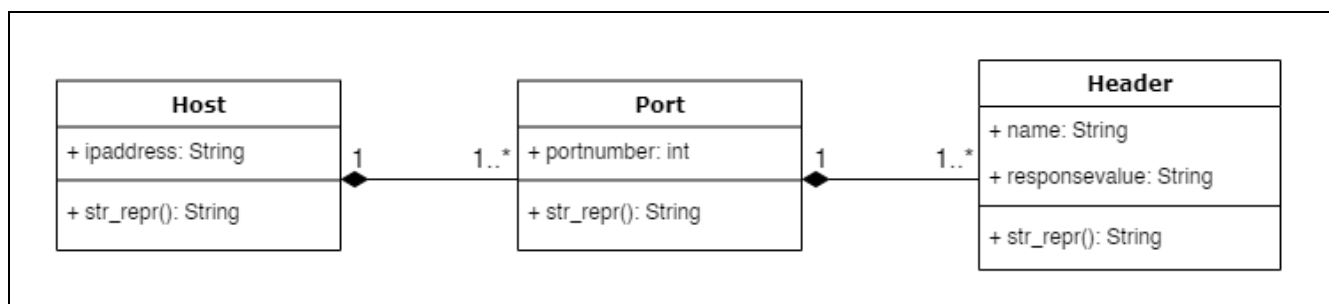


Diagrama de clases del proyecto (fase de análisis)

5.5.2 Descripción de las Clases

Nombre de la Clase
Host
Descripción
Clase que representa un host activo encontrado durante un escaneo
Responsabilidades
Representar un host activo y sus atributos para trabajar con él durante el resto de los procesos de la herramienta
Atributos Propuestos
ipaddress: Dirección IP del host encontrado que representa
Métodos Propuestos
str_repr(): Representación en texto del objeto y sus atributos para depurar el código



Nombre de la Clase
Port
Descripción
Clase que representa un puerto con un servicio web encontrado durante un escaneo
Responsabilidades
Representar un puerto con un servicio web y sus atributos para trabajar con él durante los procesos de la herramienta
Atributos Propuestos
portnumber: Número de puerto del puerto encontrado que representa
Métodos Propuestos
str_repr(): Representación en texto del objeto y sus atributos para depurar el código

Nombre de la Clase
Header
Descripción
Clase que representa una cabecera HTTP que implementa un servicio web encontrado durante un escaneo
Responsabilidades
Representar una cabecera HTTP y sus atributos para trabajar con ella durante el resto de los procesos de la herramienta
Atributos Propuestos
name: Nombre de la cabecera responsevalue: Valor de respuesta de la cabecera
Métodos Propuestos
str_repr(): Representación en texto del objeto y sus atributos para depurar el código

5.6 ASI 6: DEFINICIÓN DE INTERFACES DE USUARIO

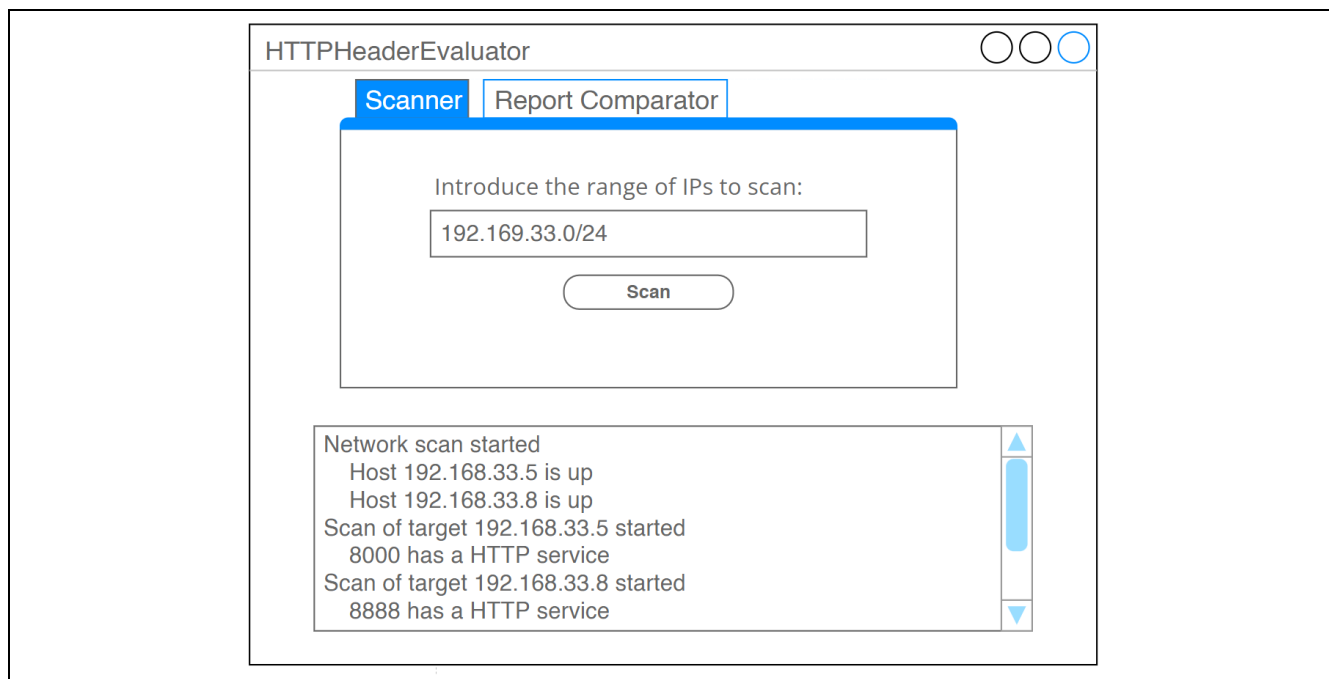
5.6.1 Descripción de la Interfaz

La interfaz de usuario consistirá en una ventana de escritorio con un menú de pestañas y un área de texto. El menú incluirá una pestaña para acceder al escáner de redes, que abrirá una subinterfaz con un campo de texto en el que el usuario podrá introducir el rango de IPs a escanear y un botón para lanzar el escáner con el rango especificado. Otra pestaña servirá para acceder al comparador de informes, a través de otra subinterfaz con dos selectores de archivos y un botón para lanzar el comparador con los archivos especificados. El área de texto no será editable y se usará para enviar al usuario mensajes del progreso de las operaciones que se estén realizando y de los problemas que

se encuentre. Esto incluye tanto, que el rango especificado en el escáner no sea válido, como que las rutas de los archivos seleccionados en el comparador sean erróneas.

5.6.2 Definición del aspecto de la Interfaz

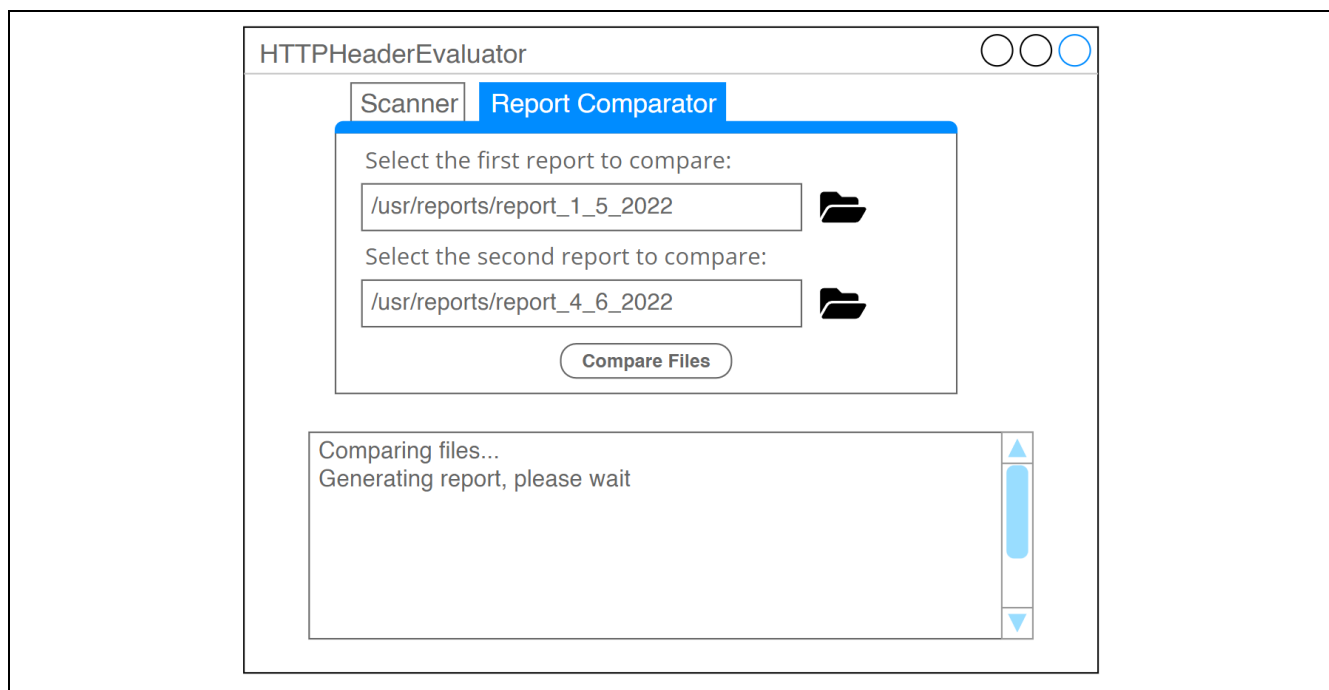
A partir de la especificación del apartado anterior, podemos diseñar las pantallas necesarias, que serían las siguientes:



The screenshot shows a window titled "HTTPHeaderEvaluator" with two tabs: "Scanner" (selected) and "Report Comparator". The "Scanner" tab contains a text input field with the value "192.169.33.0/24" and a "Scan" button. Below this is a scrollable text area displaying the following log output:

```
Network scan started
Host 192.168.33.5 is up
Host 192.168.33.8 is up
Scan of target 192.168.33.5 started
8000 has a HTTP service
Scan of target 192.168.33.8 started
8888 has a HTTP service
```

Prototipo de la interfaz del Escáner de Red



The screenshot shows the same "HTTPHeaderEvaluator" window, but with the "Report Comparator" tab selected. It contains two text input fields for selecting reports, each followed by a folder icon. The first field contains "/usr/reports/report_1_5_2022" and the second contains "/usr/reports/report_4_6_2022". A "Compare Files" button is located below these fields. The scrollable text area at the bottom displays the following status:

```
Comparing files...
Generating report, please wait
```

Prototipo de la interfaz del Comparador de Informes

5.6.3 Descripción del Comportamiento de la Interfaz

En la aplicación solo hay 3 entradas de datos:

- **Rango de direcciones IP (escáner de red):** Que se puede validar con el módulo *ipaddress* [16]. Hay tres posibles errores en esta entrada que hay que mostrar al usuario:
 - **Máscara de red incorrecta:** El valor del prefijo CIDR especificado no está entre 1 y 32. Ejemplo: 192.168.32.0/**53**
 - **Dirección incorrecta:** Uno o más de los octetos que forman la dirección IP del bloque CIDR especificado no tiene un valor entre 0 y 255. Ejemplo: 192.**351**.32.0/27
 - **Bloque CIDR incorrecto:** El número de bits reservados para la subred debería ser 0 y no es así. Ejemplo: 192.168.**255**.0/**20**
- **Informes 1 y 2 a comparar (comparador de informes):** El problema que puede encontrarse en cualquiera de estas dos entradas es que la ruta especificada no pertenezca a un archivo real, que se puede validar con la librería *os.path* [17]. Se tendría que mostrar al usuario un mensaje de error por cada ruta que no exista.

5.7 ASI 7: ESPECIFICACIÓN DEL PLAN DE PRUEBAS

5.7.1 Pruebas Unitarias

5.7.1.1 Escáner de red

En el escáner de red tenemos dos métodos a probar: *input_validation* y *scan_targets*. Para *input_validation* tenemos que comprobar que el bloque CIDR introducido por el usuario tiene un formato correcto, tal que:

- Si el valor del prefijo CIDR no está entre 1 y 32, debería saltar una excepción *NetmaskValueError*.
- Si uno o más de los valores de los octetos de la IP no está entre 0 y 255, debería saltar una excepción *AddressValueError*.
- Si en el bloque CIDR hay un número de bits reservados para la subred que deberían ser 0 y no lo son, debería saltar una excepción *ValueError*.

Y con lo que respecta a *scan_targets*, hay que probar que encuentra las máquinas correspondientes en los rangos de red en los que tienen asignada su dirección IP.

5.7.1.2 Escáner de puertos

El escáner de puertos posee tres métodos que hay que probar: *header_request*, *header_comparison*, y *scan_ports*. Para empezar, en *header_request* tendremos que comprobar que en el diccionario de cabeceras que nos devuelve están incluidas todas las cabeceras que esperamos y no más. A continuación, en *header_comparison*, habrá que confirmar que entre la lista de objetos *Header* generada se encuentran las cabeceras encontradas y con los atributos correctos. Y finalmente, en *scan_ports*, tendremos que cerciorarnos de que devuelve la lista con los puertos activos y ninguno más.

5.7.1.3 Comparador de informes

En este módulo tenemos dos métodos a probar: *input_validation* y *compare_files*. Para *input_validation* tenemos que comprobar que las rutas introducidas por el usuario son correctas, de forma que:

- Si las dos rutas no pertenecen a ningún archivo, debería devolver Falso.
- Si una ruta no pertenece a ningún archivo pero la otra sí, debería devolver Falso.
- Si las dos rutas pertenecen a un archivo, debería devolver Verdadero.
- Si una o las dos rutas están vacías, debería devolver Falso.

Mientras que en *compare_files* debemos comprobar que el método encuentra y escribe las diferencias entre los dos archivos, tal que:

- Si no hay diferencias entre los dos archivos, no debería haber ninguna línea marcada como diferente.
- Si hay líneas modificadas, debería haber dos líneas marcadas como diferentes por cada modificación (antes y después).
- Si hay líneas eliminadas, debería haber una línea marcada como diferente por cada eliminación (antes).
- Si hay líneas añadidas, debería haber una línea marcada como diferente por cada adición (después).
- Si hay líneas modificadas, eliminadas, y añadidas, debería haber el número correspondiente de líneas marcadas como diferentes.

5.7.2 Pruebas de Integración

En estas pruebas tenemos que verificar que la herramienta produce los informes que debería a partir de los distintos tipos de entradas de la interfaz de usuario. Podemos organizar estos test a partir de los casos de uso, como se ve a continuación:

Caso de Uso 1: Escanear Red	
Prueba	Resultado Esperado
Escanear un rango de red que no contiene ningún host con servicios web	El sistema no genera un informe nuevo
Prueba	Resultado Esperado
Escanear un rango de red que contiene un solo host con servicios web	El sistema genera un informe nuevo con la información esperada de las cabeceras
Prueba	Resultado Esperado
Escanear un rango de red que contiene dos hosts con servicios web	El sistema genera tres informes nuevos con la información esperada de las cabeceras

Caso de Uso 2: Comparar Informes	
Prueba	Resultado Esperado
Comparar dos rutas de uno o dos informes no existentes	El sistema no genera un informe nuevo
Prueba	Resultado Esperado
Comparar dos rutas de informes existentes pero sin diferencias	El sistema genera un informe nuevo que muestra que no hay diferencias entre los informes
Prueba	Resultado Esperado
Comparar dos rutas de informes existentes y con diferencias	El sistema genera un informe nuevo con la información esperada de las diferencias entre los informes

Capítulo 6. DISEÑO DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

DSI

6.1 DSI 1: DISEÑO DE CASOS DE USO REALES

6.1.1 Caso de Uso 1 – Escanear Red

Para el diseño de este caso de uso, suponemos que lo primero que tiene que suceder es que el usuario seleccione la vista del escáner en la interfaz principal (si no está ya abierta). Una vez hecho eso, tendrá un campo disponible en el que introducir el rango de direcciones IP en formato CIDR, y al pulsar un botón la interfaz del escáner lanzará un evento que tendrá que capturar el bucle de eventos de la interfaz principal, lanzando el escáner con la información introducida por el usuario.

El escáner validará esa información, mandando un mensaje de error de no ser correcta y parando la ejecución. Después, escaneará los hosts que se encuentren en el rango especificado, y de haber encontrado al menos uno, lanzará el escáner de puertos con todos ellos. Este escáner buscará qué puertos hospedan servicios web, y mandará peticiones HTTP a los que sí lo hagan para obtener las cabeceras que implementan. El escáner pedirá a `header_parser.py` la lista de cabeceras que deberían implementar los servicios, que creará a partir de un diccionario creado previamente con toda la información necesaria. El escáner comparará esas cabeceras con las encontradas en el servicio web para analizar cuáles están implementadas y cuáles no, lanzando el constructor de informes con los resultados obtenidos. El constructor generará los informes individuales de las máquinas escaneadas y el que ponga todos los resultados en conjunto (de haber más de una máquina), abriéndolos en el navegador del usuario en cuanto estén completos.

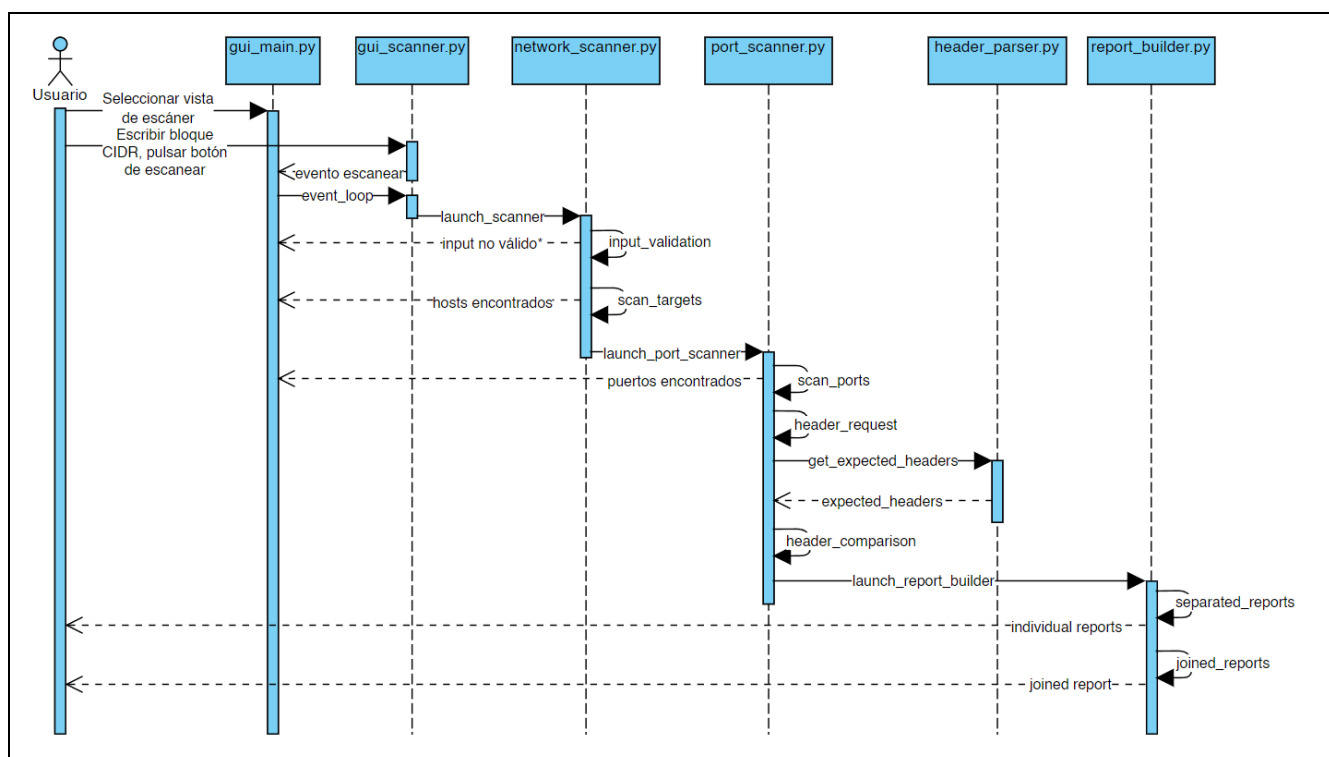


Diagrama de secuencia del Caso de Uso 1 – Escanear Red

6.1.2 Caso de Uso 2 – Comparar Informes

Al igual que en el caso de uso anterior, suponemos que primero el usuario selecciona la vista del comparador en la interfaz principal (si no está ya abierta). Tras eso, el usuario tendrá una forma de seleccionar dos informes de su sistema de archivos y un botón que al ser pulsado hará que la interfaz del comparador lance un evento que tendrá que capturar el bucle de eventos de la interfaz principal, ejecutando el comparador de informes con las rutas de archivo introducidas por el usuario.

Después de eso, el comparador validará que las rutas especificadas pertenecen a un archivo cada una, mostrando un mensaje de error de no ser así y parando su ejecución. Si las rutas son correctas, abrirá los archivos, comparará sus contenidos, y construirá un nuevo informe con las diferencias encontradas, que abrirá en el navegador del usuario una vez esté completo.

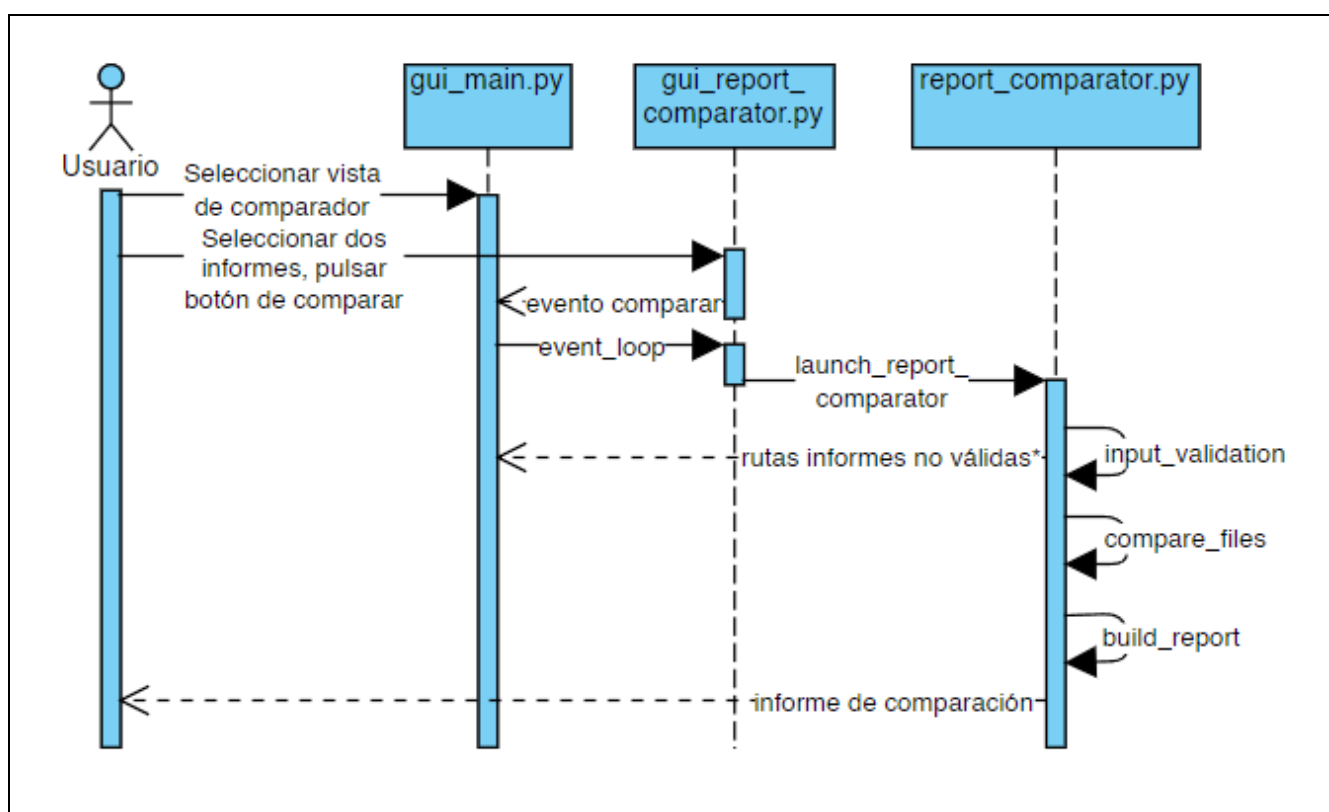


Diagrama del Caso de Uso 2 – Comparar Informes

6.2 DSI 2: DISEÑO DE CLASES

6.2.1 Diagrama de Clases

Al igual que durante la fase de análisis, el diagrama de clases sigue siendo sencillo. Lo único que cabe destacar de esta evolución es que se han añadido una serie de atributos a la clase Header, necesarios para representar toda la información que requiere la herramienta.

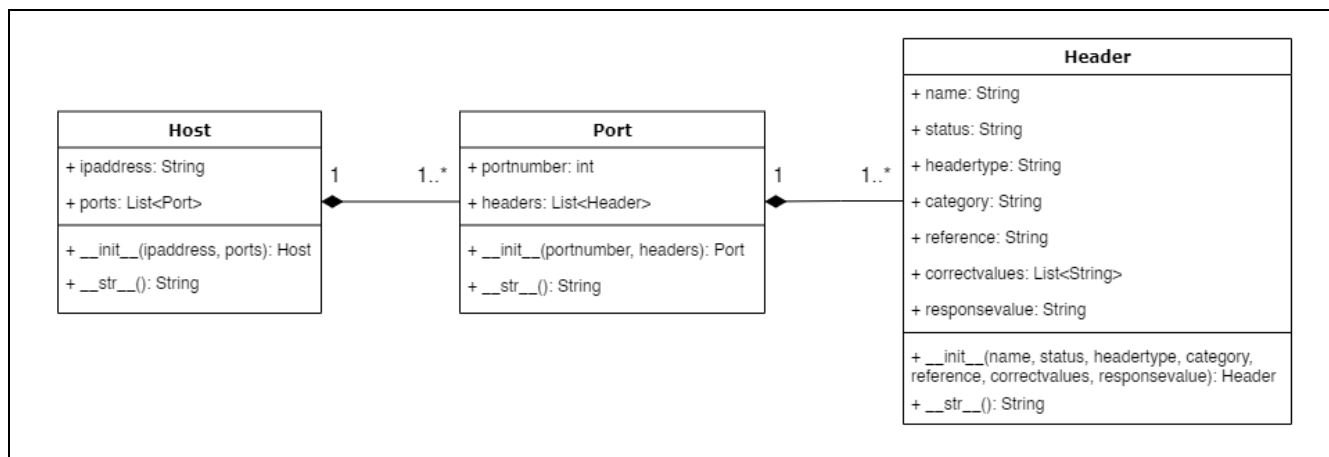


Diagrama de clases del proyecto (fase de diseño)

6.2.2 Descripción de las Clases

Nombre de la Clase
Host
Descripción
Clase que representa un host activo encontrado durante un escaneo
Responsabilidades
Representar un host activo y sus atributos para trabajar con él durante el resto de los procesos de la herramienta
Atributos Propuestos
ipaddress: Dirección IP del host encontrado que representa
<u>Nuevo atributo:</u>
ports: Lista de elementos Port que representa los puertos con servicios web encontrados
Métodos Propuestos
__str__(): Representación en texto del objeto y sus atributos para depurar el código. Nombrado de la convención de Python (originalmente str_repr())
<u>Nuevo método:</u>
init(ipaddress, ports): Constructor para instanciar la clase



Nombre de la Clase
Port
Descripción
Clase que representa un puerto con un servicio web encontrado durante un escaneo
Responsabilidades
Representar un puerto con un servicio web y sus atributos para trabajar con él durante los procesos de la herramienta
Atributos Propuestos
portnumber: Número de puerto del puerto encontrado que representa <u>Nuevo atributo:</u> headers: Lista de elementos Header que representa las cabeceras HTTP que implementa el servicio web encontrado
Métodos Propuestos
__str__(): Representación en texto del objeto y sus atributos para depurar el código. Nombrado de la convención de Python (originalmente str_repr()) <u>Nuevo método:</u> init(portnumber, headers): Constructor para instanciar la clase

Nombre de la Clase
Header
Descripción
Clase que representa una cabecera HTTP que implementa un servicio web encontrado durante un escaneo
Responsabilidades
Representar una cabecera HTTP y sus atributos para trabajar con ella durante el resto de los procesos de la herramienta
Atributos Propuestos
name: Nombre de la cabecera responsevalue: Valor de respuesta de la cabecera <u>Nuevos atributos:</u> status: Estado de implementación (implementado o no) headertype: Tipo de cabecera (estándar, deprecada, experimental) category: Categoría a la que pertenece la cabecera reference: Enlace a una web que incluya más información sobre la cabecera correctvalues: Lista blanca de valores que puede tener la cabecera para considerarse segura
Métodos Propuestos
__str__(): Representación en texto del objeto y sus atributos para depurar el código. Nombrado de la convención de Python (originalmente str_repr()) <u>Nuevo método:</u> init(name, status, headertype, category, reference, correctvalues, responsevalue): Constructor para instanciar la clase

6.3 DSI 3: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

6.3.1 DSI 3.1 Diagrama de Componentes del Sistema

En este apartado se incluye el diagrama de componentes del sistema desarrollado y una explicación de cada uno de sus elementos:

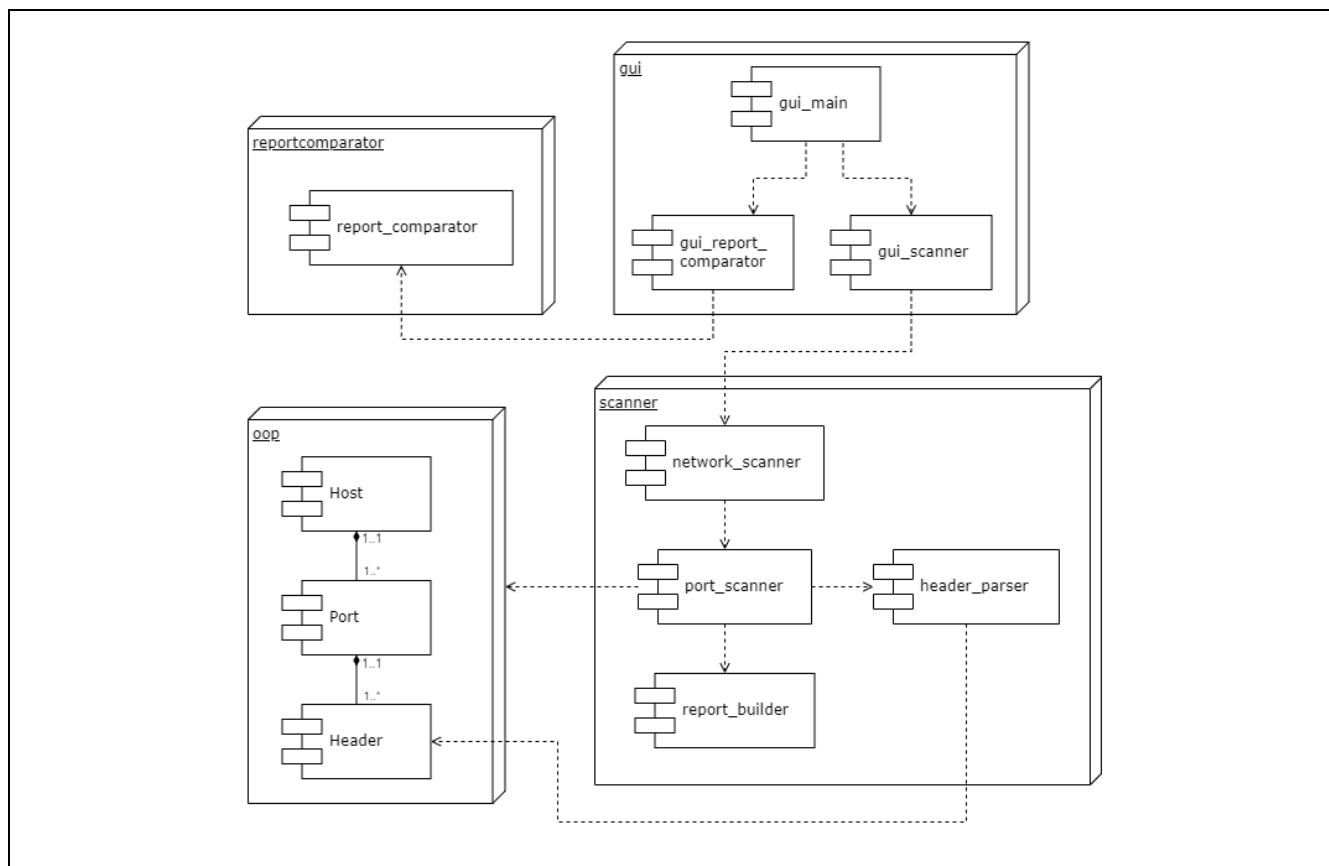


Diagrama de componentes del sistema

- **gui**
 - **gui_main**: Principal interfaz de usuario del sistema, concentra las subinterfaces de las dos funcionalidades principales y gestiona los eventos producidos en ellas.
 - **gui_report_comparator**: Subinterfaz del comparador de informes, incluye los campos necesarios para que el usuario pueda introducir la información requerida y lanza `report_comparator` con ella.
 - **gui_scanner**: Subinterfaz del escáner de redes, incluye los campos necesarios para que el usuario pueda introducir la información requerida y lanza `network_scanner` con ella.
- **reportcomparator**
 - **report_comparator**: Módulo de código que se encarga de generar un informe de comparación en formato HTML a partir de dos informes introducidos por el usuario en `gui_report_comparator`.

- **scanner**

- ***network_scanner***: Código que se encarga de encontrar hosts activos en un rango de red especificado por el usuario en *gui_scanner*. Si encuentra uno o más, llama a *port_scanner* con sus direcciones IP.
- ***port_scanner***: Escanea los puertos de las direcciones IP especificadas por *network_scanner*, usando la información encontrada y *header_parser* para generar los elementos del dominio pertinentes.
- ***header_parser***: Transforma toda la información contenida en el diccionario de cabeceras en objetos Header para su posterior tratamiento.
- ***report_builder***: Transforma todos los objetos del dominio generados por *port_scanner* en informes HTML según corresponda.

- **oop**

- ***Host***: Clase que representa un host activo.
- ***Port***: Clase que representa un puerto con un servicio web de un host activo.
- ***Header***: Clase que representa una cabecera HTTP implementada por un servicio web.

6.3.2 DSI 3.2 Revisión de la Interfaz de Usuario

La interfaz de usuario estará compuesta por una ventana que agrupará dos subinterfaces, una para el escáner de red y otra para el comparador de informes. Esta interfaz “principal” deberá ser modular, de forma que si se quisieran añadir más funcionalidades a la herramienta, se puedan incluir las nuevas subinterfaces que hicieran falta sin problema. La interfaz principal debería incluir un menú de pestañas para acceder a dichas subinterfaces y un área de texto (no editable) para notificar al usuario del progreso de las operaciones o los errores que pueda encontrar durante la ejecución de la herramienta.

Las tres interfaces se desarrollarán con la librería PySimpleGUI [13], un paquete que simplifica significativamente la construcción de interfaces de usuario en Python.

6.3.2.1 Escáner de Red

La subinterfaz del escáner de red deberá incluir un campo de texto obligatorio en el que el usuario introducirá un bloque CIDR (o alternatively, una dirección IP) y un botón para lanzar el escáner de red con la información introducida. Una vez lanzado el escáner, la interfaz se bloqueará para no permitir que se cambie de funcionalidad o se lance el escáner de nuevo sin haber acabado la ejecución previa. Además, en función del input del usuario, la interfaz mostrará uno de los siguientes mensajes:

Input	Mensaje
Bloque CIDR con mascara de red inválida	Wrong netmask input format: '{input}' is not a valid IP address/CIDR block
Bloque CIDR/dirección IP con octeto(s) erróneo(s)	Wrong address input format: '{input}' is not a valid IP address/CIDR block
Bloque CIDR con número de bits reservados para la subred distintos de 0	Wrong input format: the '{input}' network has host bits set
Bloque CIDR/dirección IP con formato correcto	Network Scan started: {#_redes_a_escanear} IPs pending to be scanned. Please, be patient - This process may take some time

En el caso de que el input del usuario entrara en una de las tres primeras categorías, tras mostrar el mensaje de error correspondiente el escáner se pararía, y la interfaz se desbloquearía, permitiendo cambiar el input o de funcionalidad. De no ser así, el escáner seguiría su curso, mostrando los siguientes mensajes, en función de su progreso:

Contexto	Mensaje
El escáner encuentra un host activo	Host {ip} is up
El escáner ha revisado el 25% de direcciones IP pendientes*	Scan Progress: 25%
El escáner ha revisado el 50% de direcciones IP pendientes*	Scan Progress: 50%
El escáner ha revisado el 75% de direcciones IP pendientes*	Scan Progress: 75%
El escáner ha revisado todas las direcciones IP y ha encontrado hosts activos	Scan Complete – Active Hosts Found: {#_hosts_activos}
El escáner ha revisado todas las direcciones IP pero no ha encontrado hosts activos	Scan Complete – No Active Hosts Found

*no tiene sentido mostrar estos mensajes cuando el número de direcciones IP a escanear es pequeño, puesto que entonces el escáner acabaría rápidamente y su objetivo es informar al usuario de que la herramienta no está bloqueada durante escaneos largos.

Si el escáner de red ha encontrado al menos un host activo, llamará al escáner de puertos con las direcciones IP correspondientes, y según progrese este, mandará estos mensajes:

Contexto	Mensaje
Empieza el escáner de puertos en un host	Scan of target {targetip} started
El escáner encuentra un servicio web en un puerto	{port} has a HTTP service
El escáner no consigue recuperar las cabeceras de un servicio web	Can't get headers from {ip}:{port}
El escáner no ha encontrado servicios web en los hosts activos	No hosts with HTTP/HTTPS services found

Finalmente, si el escáner de puertos ha encontrado al menos un servicio web del que ha podido recuperar sus cabeceras, llamará al constructor de informes, que mostrará los siguientes mensajes antes de empezar a ejecutarse:

Contexto	Mensaje
El constructor de informes tiene exactamente un host del que hacer un informe	Building scan report, please wait
El constructor de informes tiene dos o más hosts de los que hacer un informe	Building scan reports, please wait

6.3.2.2 Comparador de Informes

La subinterfaz del comparador de informes deberá incluir dos campos de texto con un selector de archivos asociado a cada uno que, tras seleccionar un informe a comparar, deberán escribir su ruta en su campo de texto correspondiente. Estos campos de texto son obligatorios y editables, pudiendo escribir la ruta del informe a comparar sin necesidad de abrir el explorador, que solo debería permitir seleccionar archivos HTML. Además, la interfaz deberá incluir un botón para lanzar el comparador con el input del usuario, que devolverá los siguientes mensajes:

Contexto	Mensaje
Se ha lanzado el comparador de informes	Opening files
El comparador no ha conseguido abrir el archivo correspondiente a la ruta de informe 1	File 1 {ruta_file1} does not exist
El comparador no ha conseguido abrir el archivo correspondiente a la ruta de informe 2	File 1 {ruta_file1} does not exist
El comparador ha conseguido abrir los dos informes a comparar satisfactoriamente	Comparing reports
El comparador está construyendo el informe	Building comparison report
El comparador ha acabado de construir el informe y va a abrirlo en el navegador	Opening result file, please wait

6.4 DSI 4: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

Para hacer las pruebas de la aplicación utilizaremos dos máquinas objetivo con direcciones IP 192.168.33.20 y 192.168.33.22, a las que llamaremos ‘máquina 1’ y ‘máquina 2’ durante este apartado. Cada una de ellas hospedará un servicio web, la primera en el puerto 8000 y la segunda en el 8008. Los detalles de ambos servicios se pueden consultar en la carpeta de scripts de servidores de prueba, ‘HTTPTestServers’, en los scripts 1 y 2 respectivamente. El primero implementa una cabecera de seguridad y dos personalizadas, mientras que el segundo solo implementa una cabecera estándar.

Si bien es cierto que no todas las pruebas requieren estas máquinas objetivo, para las pruebas tanto unitarias como de integración que sí lo hacen es fundamental, así que antes de ejecutar la suite de pruebas hay que asegurarse de que las máquinas y los servicios web correspondientes están activos.

Aparte de las máquinas objetivo, usaremos una serie de 5 informes de prueba prueba (file1route, ..., file5route), en los que los 4 últimos tienen diferencias con respecto al primero. En el segundo informe se han modificado 4 líneas, en el tercero se han eliminado 2, en el cuarto se han añadido 2, y en el quinto se han modificado 2, añadido 2, y eliminado 2.

6.4.1 Pruebas Unitarias

Para implementar las pruebas unitarias nos valdremos de unittest [18], un framework de Python que soporta todas las operaciones que podamos necesitar. Agruparemos los test por la clase que prueban, y luego los juntaremos en una suite para ejecutarlos todos de forma automática.

6.4.1.1 Escáner de red

En primer lugar tenemos el método *input_validation*, en el que tenemos que comprobar que el bloque CIDR introducido por el usuario tiene un formato correcto. Para ello, probaremos cada una de sus partes con una serie de números en los rangos válidos e inválidos, en los que se tienen que incluir valores límite y números arbitrariamente grandes o pequeños.

El prefijo CIDR ha de estar entre 1 y 32, de no ser así debería saltar una excepción `NetmaskValueError`:

Método: input_validation	
Datos de entrada	Resultado Esperado
192.168.32.0/0	NetmaskValueError
192.168.32.0/-50	NetmaskValueError
192.168.32.0/-999999	NetmaskValueError
192.168.32.0/33	NetmaskValueError
192.168.32.0/50	NetmaskValueError
192.168.32.0/999999	NetmaskValueError
0.0.0.0/1	Lista de 2,147,483,648 IP
192.168.32.0/32	Lista de 1 IP
192.168.32.0/20	Lista de 4094 IP

Todos los valores de los octetos de la IP han de estar entre 0 y 255, de no ser así debería saltar una excepción `AddressValueError`:

Método: input_validation	
Datos de entrada	Resultado Esperado
256.168.32.0/27	AddressValueError
400.168.32.0/27	AddressValueError
999999.168.32.0/27	AddressValueError
192.256.32.0/27	AddressValueError
192.400.32.0/27	AddressValueError
192.999999.32.0/27	AddressValueError
192.168.256.0/27	AddressValueError
192.168.400.0/27	AddressValueError
192.168.999999.0/27	AddressValueError
192.168.32.256/27	AddressValueError
192.168.32.400/27	AddressValueError
192.168.32.999999/27	AddressValueError

En el bloque CIDR hay un número de bits reservados para la subred que deberían ser 0. De no ser así, debería saltar una excepción `ValueError`:

Método: input_validation	
Datos de entrada	Resultado Esperado
192.168.255.0/23	ValueError
192.168.255.0/22	ValueError
192.168.255.0/13	ValueError
192.168.255.192/25	ValueError
192.168.255.192/24	ValueError
192.168.255.192/12	ValueError

Y en segundo lugar tenemos *scan_targets*, que recibe una lista de direcciones IP a escanear. Hay que probar que encuentra las máquinas en los que tienen asignada su dirección IP y que no las encuentra en los rangos en los que no. Para ello utilizaremos las máquinas 1 y 2 como objetivo, lanzando el escáner en las siguientes direcciones:

Método: scan_targets	
Datos de entrada	Resultado Esperado
[]	0 máquinas encontradas
[192.168.33.20]	1 máquina encontrada
[192.168.33.22]	1 máquina encontrada
[192.168.33.17, ..., 192.168.33.21]	1 máquina encontrada
[192.168.33.10, ..., 192.168.33.16]	0 máquinas encontradas
[192.168.33.21, ..., 192.168.33.27]	1 máquina encontrada
[192.168.33.18, ..., 192.168.33.25]	2 máquinas encontradas

6.4.1.2 Escáner de puertos

Como dijimos durante la fase de análisis, el escáner de puertos posee tres métodos que hay que probar, y para ello usaremos una máquina 1 como objetivo. El primero de ellos, *header_request*, recibe una dirección IP a escanear y un número de puerto, y devuelve un diccionario con las cabeceras que implementa el servicio, de haberlo. Tendremos que comprobar que en el diccionario de cabeceras que nos devuelve están incluidas todas las cabeceras que esperamos y no más.

Método: header_request	
Datos de entrada	Resultado Esperado
192.168.33.20, 8000	Diccionario de cabeceras con 6 elementos, siendo ellos las cabeceras 'Server', 'Date', 'Content-Type', 'Custom-header-here', 'otro-custom' y 'X-Content-Type-Options'

A continuación, en *header_comparison*, habrá que confirmar que entre la lista de objetos Header generada a partir del resultado de *header_request* se encuentran las cabeceras encontradas y con los atributos correctos, lo que quiere decir las 6 cabeceras con el atributo nombre en mayúsculas y el estado como 'DEFINED':

Método: header_comparison	
Datos de entrada	Resultado Esperado
192.168.33.20, 8000	Lista de objetos Header con 71 elementos (69 por defecto + 2 personalizados), en los que tiene que haber los 6 encontrados con los atributos especificados

Y finalmente, en *scan_ports*, que a partir de una dirección IP crea un objeto Host con su lista de objetos Port, que a su vez tienen sus listas de objetos Header, tendremos que cerciorarnos de la lista de puertos que devuelve incluye los activos y ninguno más.

Método: scan_ports	
Datos de entrada	Resultado Esperado
192.168.33.20	Objeto Host con una lista de objetos Port con 1 elemento, en el que tiene que haber la lista de objetos Header con los 6 encontrados con los atributos especificados

6.4.1.3 Comparador de informes

Durante el análisis especificamos dos métodos a probar, para el primero, *input_validation*, tenemos que comprobar que el método reconoce si las rutas introducidas por el usuario son correctas o no. Para ello, usaremos rutas a archivos no existentes y rutas a dos informes de prueba, *file1route* y *file2route*:

Método: input_validation	
Datos de entrada	Resultado Esperado
"invalid/file1/route", "invalid/file2/route"	False
file1route, "invalid/file2/route"	False
"invalid/file1/route", file2route	False
"", ""	False
file1route, ""	False
"", file2route	False
file1route, file2route	True

Para el segundo, *compare_files*, debemos comprobar que el método encuentra y escribe las diferencias entre los dos archivos. Para ello, usaremos los 5 informes de prueba (*file1route*, ..., *file5route*), y comprobaremos que el número de líneas con el tag *class='before'* o *class='after'* es el esperado:

Método: compare_files	
Datos de entrada	Resultado Esperado
file1route, file1route	0 líneas 'before' y 'after'
file1route, file2route	4 líneas 'before' y 4 'after'
file1route, file3route	2 líneas 'before' y 0 'after'
file1route, file4route	0 líneas 'before' y 2 'after'
file1route, file5route	4 líneas 'before' y 4 'after'

6.4.2 Pruebas de Integración

Las pruebas de integración, al igual de las unitarias, utilizarán unittest [18], sin embargo las funcionalidades de este framework no son suficientes para probar la interfaz de usuario. Es por eso por lo que para automatizar esa parte tendremos que valernos de una librería que permita codificar acciones de teclado y ratón. En nuestro caso esa librería será PyAutoGUI [15], aunque hay más opciones disponibles. Además, aparte de requerir las máquinas 1 y 2 como las pruebas unitarias, la interfaz de usuario tendrá que estar abierta.

Al igual que en la fase de análisis, las pruebas se organizarán a partir de los casos de uso:

6.4.2.1 Caso de Uso 1: Escanear Red

Prueba 1 - Escanear un rango de red que no contiene ningún host con servicios web

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del escáner en la interfaz.
3. Seleccionar el campo en el que se introduce el rango a escanear, eliminar su contenido, y escribir un bloque CIDR en cuyas direcciones IP no haya hosts asignados (por ejemplo, 192.168.33.0/28).
4. Pulsar el botón de escanear, y esperar a que acaben las operaciones de la herramienta.
5. Repetir el Paso 1, y comprobar que hay exactamente los mismos nombres, pues no se debería haber generado un informe nuevo.

Prueba 2 - Escanear un rango de red que contiene un solo host con servicios web

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del escáner en la interfaz.
6. Seleccionar el campo en el que se introduce el rango a escanear, eliminar su contenido, y escribir la dirección IP de la Máquina 1 (192.168.33.20).
3. Pulsar el botón de escanear, y esperar a que acaben las operaciones de la herramienta.
4. Minimizar el explorador, que se habrá abierto en cuanto se haya generado el nuevo informe.
5. Repetir el Paso 1, y comprobar que hay exactamente un nombre de archivo nuevo.
6. Comprobar que ese informe incluye la información esperada del servicio web.

Prueba 3 - Escanear un rango de red que contiene dos hosts con servicios web

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del escáner en la interfaz.
3. Seleccionar el campo en el que se introduce el rango a escanear, eliminar su contenido, y escribir un bloque CIDR en cuyas direcciones IP se encuentran las de las Máquinas 1 y 2 (por ejemplo, 192.168.33.20/29)
4. Pulsar el botón de escanear, y esperar a que acaben las operaciones de la herramienta.
5. Minimizar el explorador, que se habrá abierto en cuanto se hayan generado los 3 nuevos informes.
6. Repetir el Paso 1, y comprobar que hay exactamente 3 nombres de archivos nuevos.
7. Comprobar que en los 3 informes está incluida la información correspondiente: la de los dos servicios en uno, la de un servicio en otro, y la del otro servicio en otro.

6.4.2.2 Caso de Uso 2: Comparar Informes

Prueba 1 - Comparar dos rutas de uno o dos informes no existentes

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del comparador de informes en la interfaz.
3. Seleccionar el campo en el que se introduce la ruta del primer informe a comparar, eliminar su contenido, y escribir la ruta del informe de prueba 1 (file1route).
4. Seleccionar el campo en el que se introduce la ruta del segundo informe a comparar, eliminar su contenido, y escribir una ruta de un informe no existente.
5. Pulsar el botón de comparar, y esperar a que acaben las operaciones de la herramienta.
6. Repetir el Paso 1, y comprobar que hay exactamente los mismos nombres de archivos, porque no se debería haber generado un informe nuevo.

Prueba 2 - Comparar dos rutas de informes existentes pero sin diferencias

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del comparador de informes en la interfaz.
3. Seleccionar el campo en el que se introduce la ruta del primer informe a comparar, eliminar su contenido, y escribir la ruta del informe de prueba 1 (file1route).
4. Seleccionar el campo en el que se introduce la ruta del segundo informe a comparar, eliminar su contenido, y escribir la ruta del informe de prueba 1 (file1route)
5. Pulsar el botón de comparar, y esperar a que acaben las operaciones de la herramienta.

6. Minimizar el explorador, que se habrá abierto en cuanto se haya generado el nuevo informe.
7. Repetir el Paso 1, y comprobar que hay exactamente un nombre de archivo nuevo.
8. Comprobar que en el nuevo informe no figura ninguna línea con el tag *class='before'* o *class='after'*.

Prueba 2 - Comparar dos rutas de informes existentes y con diferencias

Pasos de la prueba:

1. Obtener los nombres de los informes en la carpeta de informes generados por la herramienta.
2. Seleccionar la ventana del comparador de informes en la interfaz.
3. Seleccionar el campo en el que se introduce la ruta del primer informe a comparar, eliminar su contenido, y escribir la ruta del informe de prueba 1 (file1route).
4. Seleccionar el campo en el que se introduce la ruta del segundo informe a comparar, eliminar su contenido, y escribir la ruta del informe de prueba 5 (file5route)
5. Pulsar el botón de comparar, y esperar a que acaben las operaciones de la herramienta.
6. Minimizar el explorador, que se habrá abierto en cuanto se haya generado el nuevo informe.
7. Repetir el Paso 1, y comprobar que hay exactamente un nombre de archivo nuevo.
8. Comprobar que en el nuevo informe hay las líneas que debería con el tag *class='before'* o *class='after'*, y las que debería sin ningún tag.

Capítulo 7. CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO



7.1 CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

7.1.1 Lenguajes de programación

El principal lenguaje del sistema es Python 3.9, que fue elegido por lo fácil que es desplegar servidores HTTP de prueba con el módulo `http.server` [19] y además por la oportunidad de explorar un lenguaje con el que tengo menos soltura comparado con otros como Java. El proyecto también tiene una parte de HTML en las plantillas de los informes, y de CSS para estilizarlas. El motivo para escoger este lenguaje de marcado fue que solo hace falta un navegador web para interpretarlo, algo que cualquier máquina en la que se despliegue la herramienta debería tener acceso.

7.1.2 Herramientas y programas usados para el desarrollo

Se ha utilizado Oracle VM VirtualBox 6.1.26 [20] y Vagrant 2.2.18 [21] para virtualizar tres máquinas en las que se han desarrollado el código y las pruebas de la herramienta. La máquina de desarrollo es un Kali Linux 2021.3 [22] con el IDE PyCharm 2021.3.3 [23], que usa la VagrantBox “kalilinux/rolling” [24]. Como máquinas objetivo se han utilizado un Ubuntu 18.04 [25] configurado a partir de un VagrantFile [26] de la asignatura de Seguridad de Sistemas Informáticos que usa la VagrantBox “hashicorp/bionic64” [27]; y un Lubuntu 20.04 [28] que usa la VagrantBox “chenhan/Lubuntu-desktop-20.04” [29]. Las 3 máquinas tienen un intérprete de Python 3.9 [30].

7.2 CSI 2: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

La ejecución de la gran mayoría de pruebas unitarias se realizó con éxito, pero sí que es cierto que hubo que cambiar algunos detalles concretos:

- En el método `input_validation` del escáner de red se esperaba que al probar con una máscara de red igual a 0 se lanzara una `NetmaskValueError`, pero resultó que la librería que valida este input considera esto como un `ValueError`, puesto que con esa máscara no hay el número de bits reservados para la subred que debería (con ninguna dirección IP). Cambiamos en esta comprobación que validara el error que le corresponde en realidad y añadimos una prueba con una máscara igual a -1 como siguiente valor límite.
- En `input_validation` también, al probar si una máscara de red igual a 32 devolvería 2147483648 direcciones IP, nos dimos cuenta de que una operación así consumía un tiempo excesivo, así que la eliminamos.
- En el método `scan_targets`, del escáner de red también, el rango de redes que se suponía que no encontraría ningún host activo incluyó por error la dirección IP de la máquina virtual que ejecutaba la herramienta, cambiando el resultado que debería salir. Cambiamos el rango de redes para asegurarnos de que esta vez no incluyera ningún host activo.

Y además, se produjeron una serie de errores que ayudaron a encontrar problemas en el código:

- En el método `header_request` del escáner de puertos, el test falló porque el servidor de pruebas devolvió dos cabeceras extra que no esperábamos (no estaban implementadas explícitamente), debido al comportamiento por defecto de los servidores HTTP de Python. Adaptamos los test con esta nueva información para arreglarlo.
- En `header_comparison`, del escáner de puertos también, el test falló en las cabeceras personalizadas debido a que el nombre de éstas no se pasaba a mayúsculas como se esperaba. Lo arreglamos cambiando el nombre de las cabeceras antes de instanciar los objetos Header.

A continuación se muestra la ejecución de los test unitarios, tras las 3 modificaciones y los 2 arreglos realizados. Se han difuminado los mensajes que produjo la herramienta durante las pruebas para facilitar la lectura:

```
===== test session starts =====
collecting ... collected 2 items

TestNetworkScannerMethods.py::TestNetworkScannerMethods::test_input_validation
TestNetworkScannerMethods.py::TestNetworkScannerMethods::test_scan_targets

===== 2 passed in 43.09s =====

Process finished with exit code 0
PASSED [ 50%]Network scan started: 1 IPs pending to be scanned
Please, be patient - This process may take some time
Network scan started: 48% IPs pending to be scanned
Please, be patient - This process may take some time
PASSED [100%]Host 192.168.33.28 is up
```

Traza de ejecución de los test unitarios del escáner de red

```
===== test session starts =====
collecting ... collected 3 items

TestPortScannerMethods.py::TestPortScannerMethods::test_header_comparison
TestPortScannerMethods.py::TestPortScannerMethods::test_header_request
TestPortScannerMethods.py::TestPortScannerMethods::test_scan_ports

===== 3 passed in 15.59s =====

Process finished with exit code 0
PASSED [ 33%]PASSED [ 66%]PASSED [100%]Scan of target 192.168.33.28 started
```

Traza de ejecución de los test unitarios del escáner de puertos

```
===== test session starts =====
collecting ... collected 2 items

TestReportComparatorMethods.py::TestReportComparatorMethods::test_compare_files
TestReportComparatorMethods.py::TestReportComparatorMethods::test_valid_input

===== 2 passed in 0.26s =====

Process finished with exit code 0
PASSED [ 50%]Comparing reports
Building comparison report
Comparing reports
Building comparison report
PASSED [100%]Loading files
```

Traza de ejecución de los test unitarios del comparador de informes

7.3 CSI 3: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

Las pruebas de integración también se ejecutaron con éxito aunque, debido a algunas particularidades de PyAutoGUI [15], implementar el diseño establecido llevó más tiempo y esfuerzo del estimado inicialmente. Los resultados obtenidos son los siguientes:

Caso de Uso 1: Escanear Red	
Prueba	Resultado Esperado
Escanear un rango de red que no contiene ningún host con servicios web	La carpeta de informes tiene los mismos informes que antes de iniciar el escaneo
	Resultado Obtenido El esperado en la especificación de la prueba
Prueba	Resultado Esperado
Escanear un rango de red que contiene un solo host con servicios web	La carpeta de informes posee un nuevo informe del host escaneado
	Resultado Obtenido El esperado en la especificación de la prueba
Prueba	Resultado Esperado
Escanear un rango de red que contiene dos hosts con servicios web	La carpeta de informes posee dos nuevos informes de los dos hosts escaneados y otro de los dos hosts en conjunto
	Resultado Obtenido El informe que debería poseer la información de los dos hosts está vacío. Solución: Se han corregido unos errores en la plantilla de hosts múltiples y en el código que genera esos informes.
Caso de Uso 2: Comparar Informes	
Prueba	Resultado Esperado
Comparar dos rutas de uno o dos informes no existentes	La carpeta de informes tiene los mismos informes que antes de iniciar el escaneo
	Resultado Obtenido El esperado en la especificación de la prueba
Prueba	Resultado Esperado
Comparar dos rutas de informes existentes pero sin diferencias	La carpeta de informes posee un nuevo informe de comparación, en el que no hay ninguna diferencia señalada
	Resultado Obtenido El esperado en la especificación de la prueba
Prueba	Resultado Esperado
Comparar dos rutas de informes existentes y con diferencias	La carpeta de informes posee un nuevo informe de comparación, en el que están las diferencias correspondientes señaladas
	Resultado Obtenido En el informe generado no se han destacado las diferencias. Solución: Se ha arreglado la generación de tags de diferencias, el nombre no correspondía al especificado en el archivo de estilo CSS

Mientras que a continuación se muestra la ejecución de las pruebas. Se tuvieron que ejecutar los métodos por separado porque al hacerlo como una suite de test el IDE no mostraba los nombres de los métodos probados:

```
===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_scanner_no_hosts

===== 1 passed in 71.01s (0:01:11) =====

Process finished with exit code 0
PASSED [100%]

===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_scanner_one_host PASSED [100%]

===== 1 passed in 69.56s (0:01:09) =====

===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_scanner_two_hosts

===== 1 passed in 72.22s (0:01:12) =====

Process finished with exit code 0
PASSED [100%]

===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_comparator_wrong_routes

===== 1 passed in 71.52s (0:01:11) =====

Process finished with exit code 0
PASSED [100%]

===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_comparator_no_differences

===== 1 passed in 71.01s (0:01:11) =====

Process finished with exit code 0
PASSED [100%]

===== test session starts =====
collecting ... collected 1 item

TestsSistema.py::TestsSistema::test_comparator_differences

===== 1 passed in 71.53s (0:01:11) =====

Process finished with exit code 0
PASSED [100%]
```

Traza de ejecución de los test de integración de la herramienta

7.4 CSI 4: ELABORACIÓN DE LOS MANUALES DE USUARIO

7.4.1 Manual de Instalación

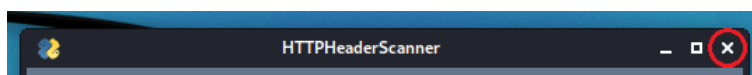
Para instalar la herramienta hay que seguir los siguientes pasos:

1. Instalar Python3.9, de no tenerlo ya instalado. En la página oficial [30] se encuentran las versiones de Windows, Mac, y Linux con los pasos correspondientes a seguir.
2. Instalar pip, de no tenerlo ya instalado (al instalar Python desde la web oficial debería incluirlo). Los pasos a seguir se encuentran en su documentación [31].
3. Instalar nmap, de no tenerlo ya instalado. Los pasos a seguir se encuentran en el apartado de descargas [32] de su web. Hay que asegurarse de que la ruta en la que se instale se añade al PATH del sistema.
4. Instalar las dependencias del código. Para ello, desde la carpeta raíz del proyecto, acceder al directorio 'instalacion', y ejecutar el comando `"pip install -r requirements.txt"`. Pip debería descargar de forma recursiva las dependencias de las dependencias, de no ser así durante la ejecución saltará un error de que no encuentra un módulo en concreto. Instalar el módulo manualmente con pip solucionará este problema.

7.4.2 Manual de Ejecución

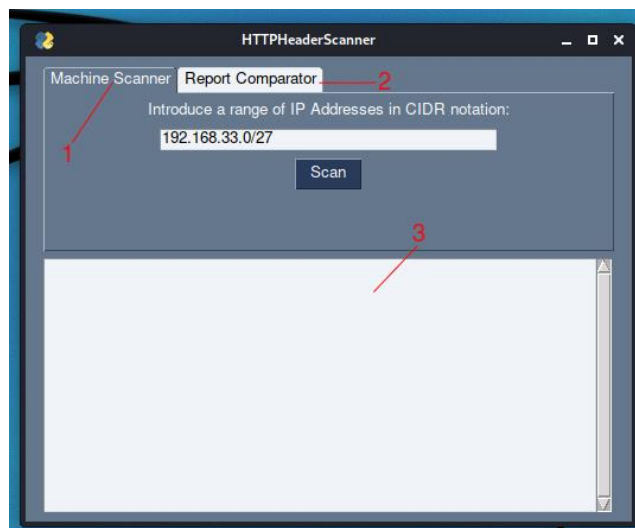
Una vez instalada la herramienta, para iniciarla solo hace falta ejecutar el script de inicio. Para ello, desde la carpeta raíz del proyecto, acceder al directorio 'http-header-evaluator' y ejecutar el script de Python 'start.py'. Dependiendo del sistema operativo, tendremos que ejecutar el comando `"python start.py"` (Windows) o `"python3 start.py"` (Linux).

En cuanto acabemos de usar la herramienta, podemos salir de ella cerrando la ventana de consola en la que hayamos ejecutado el script de inicio o haciendo click sobre la 'x' que hay en la parte de arriba a la derecha de la ventana que se abra al ejecutar el script.



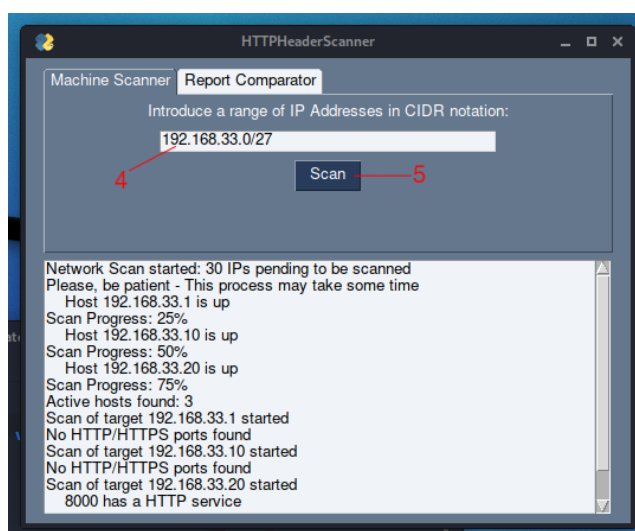
7.4.3 Manual de Usuario

Una vez que se ejecute el script de inicio, aparecerá en pantalla la siguiente ventana, que está compuesta por una serie de elementos:



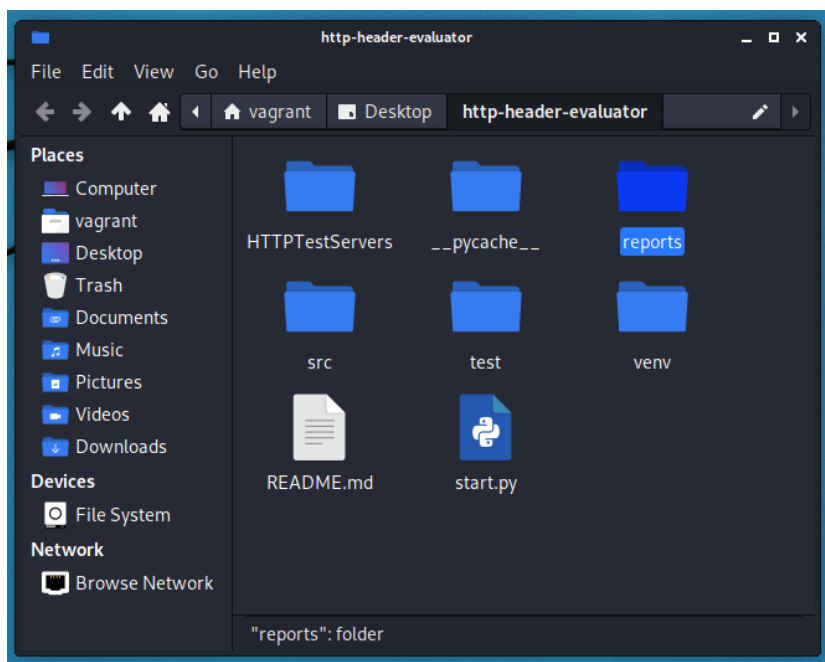
1. **Pestaña de selección del escáner:** Al hacer click en ella se muestra la pestaña del escáner (si no se está mostrando ya).
2. **Pestaña de selección del comparador de informes:** Al hacer click en ella se muestra la pestaña del comparador de informes (si no se está mostrando ya).
3. **Área de texto:** En ella se muestra el progreso de las operaciones de la herramienta o los errores que surjan durante la ejecución.

Al usar el escáner nos encontraremos las siguientes partes:



4. **Campo para introducir el rango de redes a escanear:** Se puede escribir tanto una dirección IP solamente (p.ej. 192.168.33.0) como un bloque CIDR que represente una serie de redes (p.ej. 192.168.33.0/27). Cuanto más grande sea el rango que supone el bloque CIDR, más tiempo le llevará a la herramienta completar el escaneo.
5. **Botón de escaneo:** Al hacer click en él se iniciará el escaneo con la información facilitada en 4.

Como se puede ver en la imagen, al ejecutar el escáner aparecerán en el área de texto el progreso del escaneo, los hosts activos, y los puertos que tengan servicios web. Si el escáner encuentra un host con al menos un servicio web, generará un informe sobre él; si encuentra dos o más, aparte de los informes individuales generará uno que recopile toda la información encontrada. Una vez generados, se abrirán automáticamente en el navegador web por defecto. Estos informes se pueden encontrar en la carpeta *'reports'*, en la raíz del código de la aplicación:



Estos informes están escritos en formato HTML, de forma que cualquier máquina con un navegador web pueda abrirlos. En su nombre tienen especificados la fecha en la que se realizó el escaneo, el número de cabeceras que tienen implementadas respecto a las esperadas y, en los individuales, la dirección IP de la máquina a la que corresponden. Los informes siguen una estructura como la siguiente:

HTTP Scan Report

file:///home/vagrant/Desktop/http-header-evaluator/reports/Scan_Report_192.168.33.22_2022_4_18--11:28_Headers:0of9_.html

Kali Linux Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

HTTP Scan Report

Date: 18 Apr 2022 11:28

Color	Meaning
Correct Value	Requires Manual Revision
Incorrect Value	Previous/Deleted Value
New/Added Value	

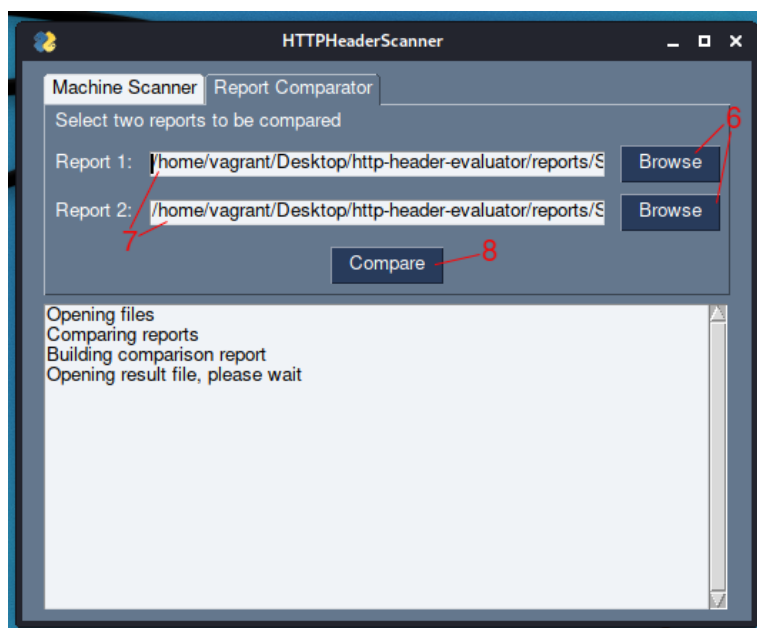
Host: 192.168.33.22

Port: 8008

Header	Status	Type	Category	Response Value	Reference
CONTENT-SECURITY-POLICY	NOT DEFINED	Standard	Security	None	Link
X-FRAME-OPTIONS	NOT DEFINED	Deprecated	Security	None	Link
X-XSS-PROTECTION	NOT DEFINED	Deprecated	Security	None	Link
PUBLIC-KEY-PINS	NOT DEFINED	Deprecated	Security	None	Link
X-CONTENT-TYPE-OPTIONS	NOT DEFINED	Standard	Security	None	Link
STRICT-TRANSPORT-SECURITY	NOT DEFINED	Standard	Security	None	Link
EXPECT-CT	NOT DEFINED	Standard	Security	None	Link
REFERRER-POLICY	NOT DEFINED	Standard	Security	None	Link
FEATURE-POLICY	NOT DEFINED	Experimental	Security	None	Link
SERVER	DEFINED	Standard	Pot. Unsafe	BaseHTTP/0.6 Python/3.8.2	Link
CONTENT-TYPE	DEFINED	Standard	Message body information	text/html	Link
DATE	DEFINED	Standard	Other	Mon, 18 Apr 2022 15:28:03 GMT	-

Primero está la fecha del escaneo, luego la leyenda de significado de los colores usados en el documento y, finalmente, las tablas con las cabeceras esperadas en los servicios web encontrados de los hosts (con su dirección IP). Las celdas de 'Status' (estado), 'Type' (tipo) y 'Response Value' (valor de respuesta) están coloreadas en función de su valor, como especifica la leyenda. Verde es un valor correcto, rojo uno incorrecto, y naranja un valor de tipo no estándar, o un valor de respuesta que la aplicación no puede determinar si es correcto o no.

Por otra parte, si queremos usar el comparador de informes hay que tener en cuenta lo siguiente:



6. **Botones para seleccionar informes:** Al pulsar sobre uno de ellos se abrirá el explorador de archivos en la carpeta de informes generados por la aplicación y se podrá seleccionar uno de los dos informes a comparar. Una vez seleccionado, se escribirá su ruta en su parte de 7 correspondiente.
7. **Campos de rutas de informes a comparar:** Campos en los que se puede introducir manualmente las rutas de los informes a comparar o modificar las escritas por los botones 6.
8. **Botón de comparación:** Al hacer click en él se iniciará la comparación con las rutas facilitadas en 7.

Al igual que en el escáner, al ejecutar el comparador se muestran mensajes del progreso, y en cuanto acaben las operaciones se abrirá en el navegador el informe generado, que se guardará en la misma carpeta de los informes del escáner.

Cabe destacar que es importante el orden en el que se seleccionen los informes, ya que la herramienta señalará las diferencias del primer informe seleccionado con respecto al segundo. Es por eso por lo que para obtener un resultado significativo habría que:

1. Seleccionar dos informes que representen una misma máquina o, en su defecto, un mismo conjunto de máquinas. Por ejemplo, comparar el resultado de dos escaneos de la máquina 192.168.33.20 o del rango de red 10.0.0.0/24 en dos momentos distintos tiene sentido, mientras que comparar un informe de la máquina 192.168.33.0 con otro de la máquina 192.255.255.255, o del rango de red 192.0.0.0/24 con el rango de red 168.0.0.0/24, no lo tiene (por lo menos para el uso intencionado de la herramienta).
2. Seleccionar como primer informe el que se haya generado antes. De esta forma, en el resultado se reflejará correctamente lo que se haya añadido, eliminado, o cambiado entre un análisis previo y otro más reciente. Si se seleccionara como primer informe el que se haya generado más tarde, los resultados serían completamente opuestos.

El resultado de este proceso tiene una estructura similar a la de los informes de escaneo, pero con los bordes de las filas con diferencias marcadas como a continuación:

HTTP Scan Report

Date	22 Apr 2022 05:28
Date	18 Apr 2022 11:31
Time Taken	0:00:30.768148
Time Taken	0:00:31.661669

Hosts
192.168.33.20
192.168.33.22

Color				
Meaning	Correct Value	Requires Manual Revision	Incorrect Value	Previous/Deleted Value
				New/Added Value

Host: 192.168.33.20

Port: 8000

Header	Status	Type	Category	Response Value	Reference
CONTENT-SECURITY-POLICY	NOT DEFINED	Standard	Security	None	Link
X-FRAME-OPTIONS	NOT DEFINED	Deprecated	Security	None	Link
X-XSS-PROTECTION	NOT DEFINED	Deprecated	Security	None	Link
PUBLIC-KEY-PINS	NOT DEFINED	Deprecated	Security	None	Link
X-CONTENT-TYPE-OPTIONS	DEFINED	Standard	Security	NOSNIFF	Link
STRICT-TRANSPORT-SECURITY	NOT DEFINED	Standard	Security	None	Link
EXPECT-CT	NOT DEFINED	Standard	Security	None	Link
REFERRER-POLICY	NOT DEFINED	Standard	Security	None	Link
FEATURE-POLICY	NOT DEFINED	Experimental	Security	None	Link
SERVER	DEFINED	Standard	Pot. Unsafe	BaseHTTP/0.6 Python/3.6.9	Link
CONTENT-TYPE	DEFINED	Standard	Message body information	text/html	Link
DATE	DEFINED	Standard	Other	Fri, 22 Apr 2022 09:28:29 GMT	-
DATE	DEFINED	Standard	Other	Mon, 18 Apr 2022 15:31:28 GMT	-
CUSTOM-HEADER-HERE	DEFINED	Custom	None	custom-content	-
OTRO-CUSTOM	DEFINED	Custom	None	custom-content-2	-

La leyenda especifica el significado de estos bordes, siendo el granate el valor anterior y el verde el nuevo. Dependiendo del contexto, estos colores significan una cosa u otra. Si una fila está coloreada de verde pero la anterior no está coloreada de granate, significa que ha sido añadida. Si está coloreada de granate pero la siguiente no está coloreada de verde es que ha sido eliminada. Y finalmente, si está coloreada de granate y la siguiente está coloreada de verde, es que ha sido modificada.

7.4.4 Manual del Programador

Diccionario de cabeceras HTTP

El análisis de cabeceras HTTP depende completamente del diccionario de cabeceras HTTP, una estructura de datos en la que se recopila toda la información sobre tipos, categorías, valores válidos, y referencias externas. Sin embargo, esta estructura se ha generado manualmente a partir de la documentación al respecto [7] de la Mozilla Developer Network (MDN), en la que se recogen los estándares más actuales y, como sabemos, a medida que pasa el tiempo estos pueden cambiar.

Es por eso por lo que, de querer usar esta herramienta en el futuro, sería recomendable revisar y actualizar esta estructura, localizada en *src/scanner/standard-response-headers-mozilla.txt*. Para añadir nuevas cabeceras solo hace falta seguir la estructura del resto de entradas del diccionario. También hay que tener en cuenta que cambiar esta estructura desencadena cambios necesarios en *scanner/HeaderParser.py*, *scanner/PortScanner.py*, *scanner/PortBuilder.py*, y *oop/Header.py*.

Generador de informes

El generador de informes (*src/scanner/ReportBuilder.py*) está profundamente relacionado con una serie de plantillas localizadas en *src/HTML*. Estas plantillas se tratan como si fueran texto, e incluyen marcadores de posición en los que se escribe información recogida por la herramienta u otras plantillas. Para modificar las plantillas hay que tener en cuenta sobre todo los marcadores de posición mencionados y su contexto, pero para el resto de cosas el generador debería ser suficientemente flexible.

Comparación de informes

Los informes se comparan mediante la librería *diff*lib [33], que genera un archivo en el que las líneas con diferencias se marcan con los caracteres '-', '+', y '?'. El comparador se limita a encontrar estas líneas marcadas y cambiar la marca por un tag de HTML con el nombre correspondiente para que se puedan apreciar visualmente las diferencias. Todos los detalles del estilo se encuentran en *src/reports/style.css*.

Capítulo 8. CONCLUSIONES Y AMPLIACIONES





8.1 CONCLUSIONES

Con este proyecto se ha creado una aplicación de escritorio que permite analizar los servicios web de una red de máquinas, comparando las cabeceras que implementen con las que cabría esperar de un servicio web seguro. La información recabada durante este proceso se sintetiza en una serie de informes, accesibles desde cualquier navegador web, que incluyen los problemas encontrados y recomendaciones para solucionarlos. Estos informes, a su vez, pueden compararse entre sí, con la herramienta desarrollada, para observar fácilmente la evolución en el tiempo de los servicios web analizados. Además, estas funcionalidades son accesibles desde una interfaz de usuario que permite usar la aplicación de forma sencilla e intuitiva.

Sin embargo, el resultado obtenido no habría sido posible sin todo el trabajo que ha requerido el proyecto. Desde la fase de análisis, en la que se determinó todo lo que debería ser capaz de hacer la herramienta; la fase de diseño, en la que se definió exhaustivamente cómo debería realizar las funciones especificadas; la fase de testeo, en la que se pudieron descubrir y solucionar errores del sistema poniendo a prueba el software desarrollado; y, por supuesto, la fase de planificación, en la que se estimaron los tiempos y recursos necesarios de las tareas que conformarían la totalidad del proyecto.

Además, cabe destacar que, una vez que se completó el proyecto, y gracias a la Universidad de Oviedo, pudimos realizar pruebas de campo en la red de la Escuela de Ingeniería Informática. Los resultados que obtuvimos ayudaron a detectar problemas en una infraestructura informática real, y consolidaron el proyecto como una herramienta valiosa de ciberseguridad.

Así que, en conclusión, gracias a este proyecto se ha desarrollado una aplicación útil para el análisis de servicios web, cubierto una necesidad importante en una institución académica de enseñanza superior, puesto a prueba todos los conocimientos adquiridos durante mi formación universitaria y, sobre todo, aprendido múltiples lecciones sobre ciberseguridad y gestión y desarrollo de proyectos software.

8.2 AMPLIACIONES

Threading

Para los sistemas que lo permitan, incluir hilos de ejecución puede mejorar significativamente el rendimiento del escaneo de redes, máquinas y puertos, permitiendo que se pueda usar la herramienta a escala mucho mayor. Sin embargo, esta tecnología tampoco aporta una funcionalidad crítica en el uso para el que está pensada la herramienta y, sobre todo, desarrollarla no es trivial en absoluto. Si no se implementa adecuadamente, puede provocar saturaciones en la red objetivo o hacer que herramientas de detección de intrusos se disparen.

Debido a la complejidad de la ampliación y el tiempo requerido para desarrollarla, se ha decidido no incluirla en el proyecto.

Configuración y personalización

Hay una serie de funcionalidades de la aplicación que podrían ganar bastante si se permitiese a los usuarios modificar el comportamiento de la herramienta. Para ello, mediante una serie de opciones de menú/interfaz podría permitirse:

- Cambiar los colores de la interfaz
- Cambiar el estilo usado en los informes de escaneo y de comparación
- Cambiar el número de informes que se generan durante el escaneo de red
- Cambiar la ruta en la que se guardan los informes de escaneo y de comparación

Dado que la naturaleza de esta ampliación se aleja sustancialmente del objetivo principal del proyecto, no se ha incluido.

Análisis de otros tipos de servicios

Aparte de servicios web, también sería posible analizar otros servicios alojados habitualmente en los puertos de una máquina, servicios como SSH o FTP, e incluir las vulnerabilidades encontradas en ellos en los informes generados por la herramienta.

Esta ampliación no se ha incluido debido a que requeriría más tiempo del disponible para realizar este trabajo.

Integración con CVE

En los servicios web que faciliten el tipo servidor que usa el servicio, podría usarse esta información para buscar en una lista CVE (*Common Vulnerabilities and Exposures*) si el servidor es vulnerable y añadir esta información relevante en los informes de escaneos. Además, existe un script de nmap (nmap-vulners [34]) que podría facilitar su desarrollo significativamente.

Aunque esta actualización de la herramienta sí que está prácticamente implementada, como todavía le faltan partes importantes para considerarla completa (su documentación correspondiente y el desarrollo de sus pruebas), figura en este apartado como posible ampliación. Sin embargo, a continuación se encuentra un ejemplo de lo que puede hacer actualmente:

CVEs Found:

CVE	Score	Reference
CVE-2021-3177	7.5	Link
CVE-2020-27619	7.5	Link
CVE-2021-3737	7.1	Link
CVE-2020-8492	7.1	Link
CVE-2020-26116	6.4	Link
MSF:ILITIES/HUAWEI-EULERO-2_0_SP2-CVE-2019-16056/	5.0	Link
CVE-2022-0391	5.0	Link
CVE-2019-9636	5.0	Link
CVE-2019-16056	5.0	Link
CVE-2018-20852	5.0	Link
CVE-2018-20406	5.0	Link
CVE-2018-1060	5.0	Link
MSF:ILITIES/REDHAT_LINUX-CVE-2016-10739/	4.6	Link
MSF:ILITIES/HUAWEI-EULERO-2_0_SP5-CVE-2016-10739/	4.6	Link
MSF:ILITIES/HUAWEI-EULERO-2_0_SP3-CVE-2016-10739/	4.6	Link
MSF:ILITIES/CENTOS_LINUX-CVE-2016-10739/	4.6	Link
MSF:ILITIES/AMAZON_LINUX-CVE-2016-10739/	4.6	Link
MSF:ILITIES/SUSE-CVE-2020-14422/	4.3	Link
MSF:ILITIES/ORACLE-SOLARIS-CVE-2020-8315/	4.3	Link
CVE-2021-28359	4.3	Link
CVE-2020-8315	4.3	Link
CVE-2020-14422	4.3	Link
CVE-2019-9947	4.3	Link
CVE-2019-9740	4.3	Link
CVE-2019-18348	4.3	Link
CVE-2019-16935	4.3	Link
CVE-2021-23336	4.0	Link
MSF:ILITIES/DEBIAN-CVE-2021-3426/	2.7	Link
CVE-2021-3426	2.7	Link

Fragmento de un informe generado por la herramienta, en el que figuran CVEs encontradas en un servidor de prueba del proyecto

Además, entre los anexos de la documentación se encuentra un informe de prueba con esta funcionalidad implementada. Este archivo se llama “informe_funcionalidad_cves.html”, se encuentra dentro de la carpeta “ejemplo_ampliacion_cves”.

APÉNDICES



PLAN DE GESTIÓN DE RIESGOS

El contenido de este apartado sirve de información adicional a lo incluido en el apartado 4.1.4 Riesgos.

Matriz de probabilidad e impacto

Esta matriz se ha usado para calcular los valores de prioridad de los riesgos en función de su probabilidad de ocurrencia y su nivel de impacto:

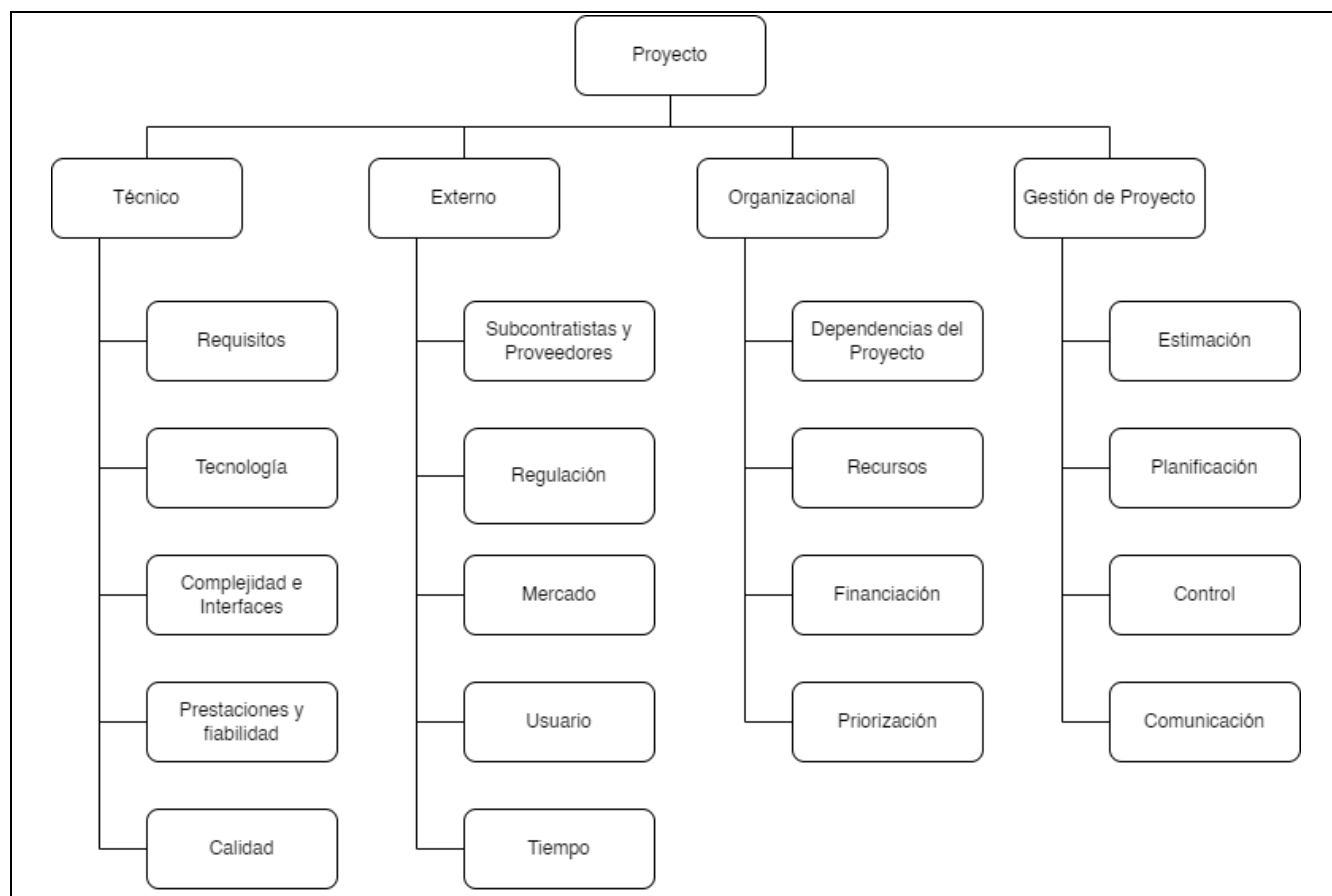
Probabilidad	Muy Alta	0,90	0,05	0,14	0,27	0,50	0,81
	Alta	0,70	0,04	0,11	0,21	0,39	0,63
	Media	0,50	0,03	0,08	0,15	0,28	0,45
	Baja	0,30	0,02	0,05	0,09	0,17	0,27
	Muy Baja	0,10	0,01	0,02	0,03	0,06	0,09
			0,05	0,15	0,30	0,55	0,90
			Inapreciable	Bajo	Medio	Alto	Crítico
			Impacto				

Matriz de Probabilidad e Impacto de riesgos del proyecto

Como no se han identificado riesgos con impactos positivos (oportunidades) en el proyecto, solo se han determinado los valores para los riesgos con impactos negativos (amenazas).

Estructura de Desglose de Riesgo. RBS

Como *Risk Breakdown Structure* del Proyecto se ha utilizado la propuesta por el PMBOK [35]:





REFERENCIAS BIBLIOGRÁFICAS

- [1] J. M. Redondo, «Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo,» 17 6 2019. [En línea]. Available: https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo.
- [2] J. Redondo, «Creación y evaluación de plantillas para trabajos de fin de grado como buena práctica docente,» Revista de Innovación y Buenas Prácticas Docentes, p. pp, 2020.
- [3] Shodan, «Shodan Search Engine,» [En línea]. Available: <https://www.shodan.io/>. [Último acceso: 1 4 2022].
- [4] G. Lyon, «Nmap: the Network Mapper - Free Security Scanner,» [En línea]. Available: <https://nmap.org/>. [Último acceso: 1 4 2022].
- [5] IONOS Cloud S.L.U, «Web services | Definición & ejemplo - IONOS,» 15 Abril 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>. [Último acceso: 10 Marzo 2022].
- [6] Mozilla Corporation, «Generalidades del protocolo HTTP - HTTP | MDN,» MDN Contributors, 9 Marzo 2022. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>. [Último acceso: 10 Marzo 2022].
- [7] Mozilla Corporation, «HTTP headers - HTTP | MDN,» MDN Contributors, 9 Marzo 2022. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Headers>. [Último acceso: 10 Marzo 2022].
- [8] «Qué es el protocolo SSL/TLS | Redalia,» [En línea]. Available: <https://www.redalia.es/ssl/protocolo-ssl/>. [Último acceso: 10 Marzo 2022].
- [9] AO Kaspersky Lab, «¿Qué es un certificado SSL y por qué es importante?,» [En línea]. Available: <https://latam.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate>. [Último acceso: 10 Marzo 2022].
- [10] D. Macario, «¿Qué son los puertos?,» 8 Noviembre 2017. [En línea]. Available: <https://appdelante.com/blog/que-son-los-puertos-networking>. [Último acceso: 10 Marzo 2022].
- [11] W. R. Stevens, TCP-IP Illustrated Volume 1 The Protocols, Addison Wesley, 1994.
- [12] «python-nmap · PyPI,» 26 10 2021. [En línea]. Available: <https://pypi.org/project/python-nmap/>. [Último acceso: 22 3 2022].
- [13] PySimpleGUI, «PySimpleGUI,» [En línea]. Available: <https://pysimplegui.readthedocs.io/en/latest/#pysimplegui-users-manual>. [Último acceso: 30 3 2022].
- [14] «Requests: HTTP for Humans™ — Requests 2.27.1 documentation,» [En línea]. Available: <https://docs.python-requests.org/en/latest/>. [Último acceso: 30 3 2022].
- [15] Sweigart, «PyAutoGUI · PyPI,» [En línea]. Available: <https://pypi.org/project/PyAutoGUI/>. [Último acceso: 2022 4 19].



- [16] Python Software Foundation, «ipaddress — IPv4/IPv6 manipulation library — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/ipaddress.html>. [Último acceso: 30 3 2022].
- [17] Python Software Foundation, «os.path — Common pathname manipulations — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/os.path.html>. [Último acceso: 30 3 2022].
- [18] Python Software Foundation, «unittest — Unit testing framework — Python 3.9.12 documentation,» [En línea]. Available: <https://docs.python.org/3.9/library/unittest.html>. [Último acceso: 2022 4 19].
- [19] Python Software Foundation, «http.server — HTTP servers — Python 3.10.4 documentation,» [En línea]. Available: <https://docs.python.org/3/library/http.server.html>. [Último acceso: 11 4 2022].
- [20] Oracle, «Oracle VM VirtualBox,» [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: 8 4 2022].
- [21] HashiCorp, «Vagrant by HashiCorp,» [En línea]. Available: <https://www.vagrantup.com/>. [Último acceso: 8 4 2022].
- [22] OffSec Services Limited, «Kali Linux 2021.3 Release (OpenSSL, Kali-Tools, Kali Live VM Support, Kali NetHunter Smartwatch) | Kali Linux Blog,» [En línea]. Available: <https://www.kali.org/blog/kali-linux-2021-3-release/>. [Último acceso: 11 4 2022].
- [23] JetBrains s.r.o., «Otras versiones: PyCharm,» [En línea]. Available: <https://www.jetbrains.com/es-es/pycharm/download/other.html>. [Último acceso: 11 4 2022].
- [24] HashiCorp, «Vagrant box kalilinux/rolling - Vagrant Cloud,» [En línea]. Available: <https://app.vagrantup.com/kalilinux/boxes/rolling>. [Último acceso: 8 4 2022].
- [25] Canonical Ltd., «Ubuntu 18.04.6 LTS (Bionic Beaver),» [En línea]. Available: <https://releases.ubuntu.com/18.04/>. [Último acceso: 11 4 2022].
- [26] J. M. R. López, «SSI_Infraestructure_Automation_Materials/Vagrantfile at main · jose-r-lopez/SSI_Infraestructure_Automation_Materials,» 16 11 2021. [En línea]. Available: https://github.com/jose-r-lopez/SSI_Infraestructure_Automation_Materials/blob/main/Vagrantfiles/machines/debian/ubuntu_cmd/Vagrantfile. [Último acceso: 4 8 2022].
- [27] HashiCorp, «Vagrant box hashicorp/bionic64 - Vagrant Cloud,» [En línea]. Available: <https://app.vagrantup.com/hashicorp/boxes/bionic64>. [Último acceso: 8 4 2022].
- [28] Canonical Ltd., «Downloads — Ubuntu,» [En línea]. Available: <https://ubuntu.me/downloads/>. [Último acceso: 11 4 2022].
- [29] «Vagrant box chenhan/lubuntu-desktop-20.04 - Vagrant Cloud,» [En línea]. Available: <https://app.vagrantup.com/chenhan/boxes/lubuntu-desktop-20.04>. [Último acceso: 8 4 2022].
- [30] Python Software Foundation, «Python Release Python 3.9.12 | Python.org,» [En línea]. Available: <https://www.python.org/downloads/release/python-3912/>. [Último acceso: 8 4 2022].



- [31] «Installation - pip documentation v22.0.4,» [En línea]. Available: <https://pip.pypa.io/en/stable/installation/#supported-methods>. [Último acceso: 2022 4 27].
- [32] G. Lyon, «Download the Free Nmap Security Scanner for Linux/Mac/Windows,» [En línea]. Available: <https://nmap.org/download.html>. [Último acceso: 28 4 2022].
- [33] Python Software Foundation, «difflib — Helpers for computing deltas — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/difflib.html>. [Último acceso: 30 3 2022].
- [34] g. A. v. D. com, «vulnersCom/nmap-vulners: NSE script based on Vulners.com API,» [En línea]. Available: <https://github.com/vulnersCom/nmap-vulners>. [Último acceso: 26 5 2022].
- [35] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide) - Fifth Edition, Newtown Square, Pennsylvania: Project Management Institute, Inc, 2013.
- [36] Python Software Foundation, «ast — Abstract Syntax Trees — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/ast.html>. [Último acceso: 30 3 2022].
- [37] Python Software Foundation, «warnings — Warning control — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/warnings.html>. [Último acceso: 30 3 2022].
- [38] Python Software Foundation, «datetime — Basic date and time types — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/datetime.html>. [Último acceso: 30 3 2022].
- [39] Python Software Foundation, «webbrowser — Convenient Web-browser controller — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/webbrowser.html>. [Último acceso: 30 3 2022].
- [40] Python Software Foundation, «re — Regular expression operations — Python 3.9.12 documentation,» 29 3 2022. [En línea]. Available: <https://docs.python.org/3.9/library/re.html>. [Último acceso: 30 3 2022].

CONTENIDO ENTREGADO EN LOS ANEXOS

Contenidos

Directorio	Contenido
./	Contiene un fichero README.TXT explicando toda esta estructura.
./http-header-evaluator	Contiene toda la estructura de directorios del proyecto para desarrollo.
./instalacion	Ficheros utilizados para la instalación del proyecto (requirements.txt) e instrucciones a seguir para instalar la herramienta (instrucciones.txt).
./documentacion	Contiene la documentación asociada al proyecto en PDF (documentacion_proyecto.pdf) y una carpeta (anexos) que contiene: <ul style="list-style-type: none">- Un archivo de MS Project de la planificación del proyecto (planificacion_proyecto.mpp).- Una carpeta (informes_red_escuela) con los informes obtenidos durante el escaneo de la red de la Escuela.- Una carpeta (ejemplo_ampliacion_cves) con un informe de prueba (informe_funcionalidad_cves.html) generado por la versión beta de la herramienta que incluye la ampliación 'Integración con CVEs'.

Estructura general del fichero anexo entregado

Estructura de Directorios de "desarrollo"

Directorio	Contenido
./http-header-evaluator	Contiene los ficheros de proyecto del IDE utilizado.
./HTTPTestServers	Scripts de Python para lanzar servidores de prueba.
./reports	Carpeta donde se guardan los informes generados por la herramienta y su archivo de estilo (style.css).
./src	Ficheros de código fuente.
./test	Directorio base para todos los ficheros utilizados en la automatización del proceso de prueba.
./start.py	Script de Python para lanzar la aplicación una vez instaladas las dependencias.

Estructura de la carpeta "desarrollo" del fichero anexo entregado