Facultad 4

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Implementación de variables relacionadas con la seguridad para el Evaluador del Monitor de Sitios Web cubano.

Autor: José Luis Domínguez Echevarría.

Tutores: Ing. Ibelis Gutiérrez Oliva.
M. Sc. Tatiana Leyva Estrada.

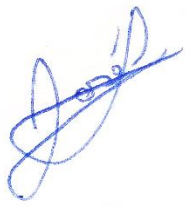
La Habana, Noviembre de 2023

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título **"*Implementación de variables relacionadas con la seguridad para el Evaluador del Monitor de Sitios Web cubano*"** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 27 días del mes de noviembre del año 2023.

José Luis Domínguez Echevarría



Firma del Autor

Ing. Ibelis Gutiérrez Oliva

Firma del Tutor

M. Sc. Tatiana Leyva Estrada

Tatiana Leyva Estrada

Firmado digitalmente por Tatiana Leyva Estrada
DN: OID.0.9.2342.19200300.100.1.1=tlestrada, CN=Tatiana Leyva Estrada, SERIALNUMBER=T110411, G=Tatiana, SN=Leyva Estrada, C=CU
Razón: Soy el autor de este documento
Ubicación:
Fecha: 2023.11.22 11:24:03+01'00'
Foxit PDF Reader Versión: 2023.2.0

Firma del Tutor

DEDICATORIA

A mis padres y hermana por su apoyo y amor incondicional en cada etapa de mi vida, dejándome saber en todo momento lo orgullosos que están de mí, dándome fuerzas para lograr mis metas.

A mi compañera de vida, mi mujer, mi amor; por hacer suyo en la mayoría de ocasiones todo el sacrificio y trabajo que ha significado esta tesis, demostrándome que el amor también es luchar juntos.

A mis compañeros y amistades de carrera Jean, Yilber, Josué, Elier, Felipe, Isabel, Kevin y, al que considero como un hermano para mí y además el coautor de esta tesis, Osmany, por no solo apoyarme y servirme de guía, sino también por cada momento de jodedera y desesteres, ya que como diría el: “Todavía hay tiempo y lo importante es divertirse”.

A mis hermanos de azul, ya que gracias a cada uno de esos momentos de fraternidad, alegría y cariño incondicional, originados en ese pequeño gran lugar llamado Humboldt 7, soy en gran medida la persona de hoy en día.

A mis amistades de tertulias y café de madrugada Jesus, Jorgito y Lavielle sin los cuales no sería capaz de llegar a esas grandes ideas para cambiar el mundo.

A mis tutoras Ibelis y Tatiana por cada una de esas horas empleadas para lograr que esta tesis tenga la calidad que se merece y por estar siempre al pendiente de mí.

A TODOS MUCHAS GRACIAS.

Resumen.

La seguridad en internet abarca todas las medidas de precaución tomadas para proteger los componentes de la red, como la infraestructura y la información. En el ámbito del intercambio de información en internet, existe el riesgo de diversos peligros comunes que afectan tanto a los usuarios como a los sitios web, como el robo de datos personales o bancarios, la presencia de virus y el robo de identidad. Con el propósito de mejorar el posicionamiento de los sitios web y brindar recomendaciones para fortalecer la seguridad de estos sitios, se han desarrollado variables de seguridad. Estas variables permiten aumentar la visibilidad de los contenidos de calidad en internet. Para implementar esta solución, se ha seguido la metodología Programación Extrema (XP) y se ha seleccionado como tecnologías clave el *framework Spring,* mediante *Spring Boot,* para programación en Java. Además, se utiliza la herramienta de modelado *Visual Paradigm*. Los procesos implementados presentan características y funcionalidades que contribuyen a mejorar la evaluación de la seguridad para el posicionamiento de los sitios web. Estos procesos brindan una evaluación que varía de 0 (incorrecto) a 1 (mejorable) y 2 (correcto). Esta evaluación se muestra en el Monitor de Sitios Web cubano y en el Seowebmas.

Palabras Clave: SEO, evaluador, seguridad, posicionamiento, sitios web, Java.

Abstract.

Internet security encompasses all precautionary measures taken to protect network components, such as infrastructure and information. In the realm of internet information exchange, there is a risk of various common dangers that affect both users and websites, such as personal or financial data theft, the presence of viruses, and identity theft. To improve website positioning and provide recommendations to enhance their security, security variables have been developed. These variables help increase the visibility of high-quality content on the internet. The XP (Extreme Programming) methodology has been followed to implement this solution, with Spring framework, through Spring Boot, chosen as a key technology for Java programming. Additionally, MySQL is used as the database management system, and Visual Paradigm as the modeling tool. The implemented processes possess characteristics and functionalities that contribute to improving website positioning. These processes provide an evaluation ranging from 0 (incorrect) to 0.5 (improvable) and 1 (correct). This evaluation is displayed in the "Monitor de Sitios Web cubano" and "Seowebmas".

Keywords: SEO, evaluator, security, positioning, websites, Java

Índice.

Índice de Tablas.

Índice de Figuras.

Introducción.

Las Tecnologías de la Información y la Comunicación (TIC) han transformado de manera vertiginosa la vida cotidiana y social de los seres humanos, algunos ejemplos están en el uso de los teléfonos móviles, los computadores, el internet y sus herramientas de comunicación. Estas transformaciones han ido permeando los ámbitos profesionales y educativos para facilitar nuestros desempeños en varias áreas, una de ellas tiene que ver con el acceso a la información (Arbeláez Gómez, 2014).

Cuando se necesita realizar una búsqueda, incluso en el tema más sencillo, Internet es la primera opción a tener en cuenta y no en una biblioteca tradicional. Es que Internet, es como una gran biblioteca, con múltiples departamentos especializados en diferentes materias. Internet es indiscutiblemente un medio de publicación rápido, libre de arbitraje, requisitos y normas, con cobertura internacional, que hace de ella el soporte preferido. Además el hecho de que muchos documentos que se editan en soporte impreso tienen versiones en Internet precisamente a favor de su acceso y visibilidad. Estos aspectos explican por sí solos el hecho de que la cantidad de información disponible en el web sea abrumadora. Afortunadamente, a la par del crecimiento de Internet se han desarrollado y perfeccionado los motores de búsqueda, dirigidos a facilitar la navegación y el hallazgo de la información necesaria (Torres Pombert, 2003).

Evidentemente, si se comparan los motores de búsqueda de hace unos años atrás con los actuales será fácil percatarse de que la cantidad de información procesada en sus bases de datos es mucho mayor, debido precisamente a que la información en la red se multiplica a diario. Por otra parte, se estima que, mientras en 1995, apenas existía una docena de motores de búsqueda, hoy se calculan en alrededor de 2000, cada uno con características diferentes, facilidades particulares, formas de funcionamiento e interfaz propia. Si bien es cierto que en el inicio los motores de búsqueda, la preocupación de los navegantes era encontrar alguno cuyo *host* estuviera disponible en el momento en que fuera a hacerse uso de él o simplemente

saber cuál realizaría la búsqueda de manera más fácil, en la actualidad el primer problema está en identificar, seleccionar y decidirse por uno de ellos.

El primer problema que enfrentan los buscadores para ordenar sus resultados es que no existe una fórmula matemáticamente precisa que permita el "mejor" ordenamiento. La cuestión más difícil es que tienen que asumir no solo el ordenamiento de una búsqueda muy precisa de 20 registros como resultado, lograda por una perfecta combinación de términos y frases, sino que también deben ser capaces de ordenar una búsqueda realizada por una sola palabra que arroje millones de registros (Torres Pombert, 2003).

Uno de los factores que más preocupa a cualquier persona o *webmaster* que tenga un portal web en internet, es el tráfico que reciba dicho proyecto. La mejor forma de incrementar este tráfico tan deseado, es mediante el posicionamiento web o SEO (*Search Engine Optimization*). Varios autores se han pronunciado por este concepto teniendo en cuenta su relevancia hoy en día, entre ellos:

El SEO es un aspecto imprescindible hoy en día. Cualquier institución, empresa o negocio que se encuentre en internet debe cuidar su posicionamiento SEO para aparecer en las primeras posiciones de los motores de búsquedas. Tener una web bien posicionada hará que ésta tenga más tráfico web, por lo tanto, mayor visibilidad respecto a la competencia y mayor probabilidad de atraer a nuevos clientes, pues tener un buen posicionamiento web en el medio online es como tener una buena localización en el mundo real (Botton, 2018).

Estas estrategias han dado lugar a dos grandes ramas del SEO, SEO *On Page*, el cual se centra en las acciones para optimizar los contenidos de la propia página web y SEO *Off Page*, este centrándose en las acciones para conseguir enlaces de entrada (Gonzalez Garay et al., 2021). SEO interno, también conocido como SEO *On Site* o SEO *On Page*, resalta que se trata de optimizar aquellos aspectos internos del diseño y estructura de la página web que se quiere posicionar.

La herramienta Monitor de Sitios Web cubano tiene como objetivo evaluar el nivel de implementación de las técnicas de posicionamiento web. Este macroproyecto se

encarga de optimizar el sistema web que evalúa y prepararlo para que alcance una mayor autoridad y relevancia en los resultados orgánicos de los motores de búsquedas. Actualmente analiza un total de treinta y seis variables SEO relacionadas con la comunicación de los sistemas web con los motores de búsqueda y las redes sociales, analizando temas de usabilidad, rendimiento, seguridad y optimización de la infraestructura donde están hospedados estos sistemas.

Sin embargo temas tan importantes como el nivel de seguridad implementado en un sistema web se tienen en cuenta de manera muy superficial, ya que solo se evalúan los parámetros relacionados con la divulgación de direcciones de correo electrónico, si se utiliza el protocolo seguro HTTPS, la protección contra ataques de tipo *Cross-Site (X-XSS-Protection)* y *Clickjacking*; mientras que no se tienen en cuenta otros parámetros como la divulgación de la tecnología con la que se desarrolló el sistema, la seguridad en las *cookies* y diversos tipos de ataques de inyección de código. Que un sistema web no cuente con las condiciones necesarias para enfrentar los ataques más comunes que se pudieran realizar pone en peligro tanto a los usuarios que los visitan como a los propios objetivos, estatus moral, aspectos económicos e incluso la ética del sitio web, afectando a la institución o persona responsable del medio digital.

A partir de lo anteriormente descrito se plantea el siguiente **problema de investigación**:
¿Cómo lograr que el evaluador realice un mejor diagnóstico asociado a los temas de seguridad de los sistemas web y brinde recomendaciones más acertadas para que los *webmaster* mejoren el posicionamiento de sus sitios?

Como **objeto de estudio** se establece el proceso de verificación y evaluación de la seguridad de los sistemas web.

Objetivo general: Implementar variables que permitan el diagnóstico de la gestión de la seguridad de sistemas web al utilizar el Monitor de Sitios Web cubano y así contribuir a las recomendaciones de mejoras que se les brindan a los *webmaster*.

Se define como **campo de acción** los indicadores de las variables para el Monitor de Sitios Web cubano que permitan la evaluación de la seguridad de los sistemas web.

Preguntas científicas:

- ¿Cuáles son los fundamentos teóricos-metodológicos y conceptos asociados a la verificación de la seguridad de sistemas web?
- ¿Cuáles son las características con las que deben contar los servicios del Monitor de Sitios Web cubano para la evaluación de seguridad en sistemas web?
- ¿Qué resultados se obtendrán al concluir el desarrollo de los servicios para el Monitor de Sitios Web cubano?

Tareas de investigación:

- Analizar bibliografía sobre conceptos asociados a la seguridad en los sistemas web para una mayor profundización en el tema.
- Identificar herramientas e indicadores para la gestión de la seguridad en sistemas web.
- Identificar variables asociadas a la seguridad implementadas en el Evaluador e identificación de carencias para centrar el trabajo en los puntos de mayor vulnerabilidad.
- Seleccionar tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
- Implementar la solución del trabajo de diploma para contribuir al perfeccionamiento del Evaluador del Monitor de Sitios Web cubano.
- Realizar pruebas para validar el correcto funcionamiento de las variables desarrolladas.

Métodos científicos, técnicas e instrumentos para la recogida de datos:

Métodos teóricos:

- Analítico-Sintético: Este método permite la recopilación de información referente a las variables de mayor relevancia para la evaluación de la seguridad en los sistemas web durante la realización del estudio del estado del arte para el desarrollo del trabajo mediante la revisión de documentos y artículos relacionados con el proceso de gestión de la seguridad de los sitios web. Además del análisis de las diferentes herramientas, metodologías y tecnologías a utilizar en el desarrollo del sistema.
- Modelación: Se utiliza para crear el proceso de diseño mediante la abstracción de sus elementos fundamentales (tareas, artefactos, guías técnicas y roles) utilizando un lenguaje de modelado de procesos.

Métodos empíricos:

- Entrevista: Como técnica de recopilación de información, posibilita entender como evaluar la seguridad de un sitio web y obtener, según la opinión de expertos, cuáles serían las variables de mayor importancia para dicha evaluación. Consultar la entrevista realizada en los anexos.
- Observación: Se utiliza para estudiar más de cerca y obtener información detallada acerca del proceso de evaluación de las variables en la aplicación del Evaluador. Consultar la guía de observaciones realizada en los anexos.
- Análisis documental: Se emplea con el fin de realizar un estudio para entender el funcionamiento del Evaluador del Monitor de Sitios Web cubano.

Estructura del documento:

El presente trabajo de diploma tendrá la siguiente estructura: Introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

Capítulo 1: Fundamentación teórico-metodológica de la evaluación de la seguridad en sistemas web. Se abordan temas como posicionamiento web, amenazas a la seguridad de los sitios, sistemas homólogos, metodologías y herramientas utilizadas para el desarrollo de la investigación.

Capítulo 2: Aspectos relacionados con la implementación de variables de seguridad para el Monitor de Sitios Web cubano. Se plantea la propuesta de solución al problema de investigación y se generan los artefactos correspondientes a la

metodología seleccionada para el desarrollo de dicha solución.

Capítulo 3: Resultados del desarrollo de las variables de seguridad para el Monitor de Sitios Web cubano. Se desarrollan los artefactos y pruebas definidas para la validación de la propuesta de solución.

Capítulo 1: Fundamentación teórico-metodológica de la evaluación de la seguridad en sistemas web.

En el presente capítulo se expone conceptos asociados al problema de investigación. Además, se describen los aspectos fundamentales relacionados con la gestión de la seguridad y los efectos de la misma en el posicionamiento web. Se realiza el estudio de los sistemas homólogos más significativos dentro del campo de la evaluación de los sistemas web y se describen las herramientas, técnicas y metodología seleccionada para el desarrollo de la solución.

1.1 Conceptos asociados al problema.

Los motores de búsqueda siguen los enlaces de una página a otra, y el índice del contenido encontrado. Al realizar una búsqueda, el motor de búsqueda muestra el contenido indexado en base a su tabla de contenidos organizándolos guiados por las reglas de su algoritmo. El cumplimiento de las directrices de los buscadores con exactitud no implica el correcto posicionamiento del proyecto asociado en los mejores resultados, mientras que la ausencia de una penalización por parte del algoritmo del buscador si tiene repercusión en dicho posicionamiento.

Los motores de búsqueda dan algunas pautas para el SEO, pero los grandes motores de búsqueda mantienen el resultado de la clasificación como secreto comercial. El SEO combina las directrices oficiales del motor de búsqueda, el conocimiento empírico y el conocimiento teórico de los artículos científicos o patentes (*SEO - Glosario de MDN Web Docs*, s. f.).

1.1.1 Definición de posicionamiento web.

SEO (*Search Engine Optimization*) también conocido como posicionamiento web, es el proceso de hacer un sitio web más visible en los resultados de búsqueda o mejorar el *ranking* de búsqueda. Tiene como objetivo lograr que un sitio aparezca entre los primeros resultados de búsqueda de forma orgánica/natural, es decir, sin necesidad de pagar. Dichos resultados se activan mediante la indexación y el rastreo que realizan las “arañas web” (el robot de Google) al recorrer billones de

opciones que podrían resolver la necesidad del usuario. Se debe orientar el posicionamiento a la mayor cantidad de buscadores posibles, no solo a Google, pese a ser el más empleado, también se debe obtener un buen posicionamiento SEO en Bing, Yahoo, entre otros (*SEO - Glosario de MDN Web Docs*, s. f.).

1.1.2 Métodos de posicionamiento SEO.

Existen dos métodos de posicionamiento:

- **SEO *On Page***: Conjunto de factores internos que existen en la web a posicionar que influyen en los resultados de búsqueda. Estos factores son tales como título, descripciones, contenido y calidad del mismo, densidad de términos y correcta estructuración de contenidos (*SEO - Glosario de MDN Web Docs*, s. f.).
- **SEO *Off Page***: Conjunto de factores externos que influyen en los resultados de búsqueda. Estos factores son la experiencia del usuario en la web a posicionar, tráfico y sobre todo los enlaces externos que apuntan a la web (*SEO - Glosario de MDN Web Docs*, s. f.).

1.2 Gestión de la seguridad de los sistemas web.

La seguridad en los sistemas web tiene el objetivo de minimizar los riesgos asociados al acceso y utilización de determinado sistema de forma no autorizada y en general malintencionada, define las características, condiciones, normas, procedimientos, métodos y técnicas de sistemas de procesamiento de datos y su almacenamiento, para garantizar su confidencialidad, integridad y disponibilidad con el objetivo de conseguir un sistema de información (o informático) seguro y confiable (Niño Benitez & Silega Martínez, 2018).

Frecuentemente los sistemas web dejan de estar disponibles debido a ataques de denegación de servicio, o presentan información modificada (y con frecuencia dañada) en sus páginas de inicio. En otros casos de alto nivel, millones de contraseñas, direcciones de correo electrónico y detalles de tarjetas de crédito han sido filtrados al dominio público, exponiendo a los usuarios del sistema web tanto a bochorno personal como a riesgo financiero (*Web Security*, s. f.).

La seguridad de un sistema web eficaz requiere de esfuerzos de diseño a lo largo de la totalidad del sitio web: en la aplicación web, en la configuración del servidor web, en las políticas para crear y renovar contraseñas, y en el código del lado cliente. Mediante el uso de un *framework* web de lado servidor, pueden quedar habilitados por defecto mecanismos de defensas robustos y bien pensados contra gran cantidad de los ataques más comunes. Otros ataques pueden mitigarse por medio de la configuración del servidor web, por ejemplo habilitando HTTPS (*Web Security*, s. f.).

1.2.1 Amenazas contra la seguridad de los sitios web.

En el siguiente epígrafe se exponen una serie de ataques que constituyen grandes amenazas para los sistemas web carentes de las implementaciones de los respectivos parámetros de seguridad, con el objetivo de determinar las variables de mayor valor para la mitigación de las mismas.

Cross-Site Scripting (XSS).

XSS es un término que se usa para describir una clase de ataques que permiten al atacante inyectar scripts de lado cliente, a través del sitio web, hasta los exploradores de otros usuarios. Como el código inyectado va del servidor del sitio al explorador, se supone de confianza, y de aquí que pueda hacer cosas como enviar al atacante la *cookie* de autorización al sitio del usuario. Una vez que el atacante tiene la *cookie* pueden iniciar sesión en el sitio como si fuera el verdadero usuario y hacer cualquier cosa que pueda hacer éste. Dependiendo de qué sitio sea, esto podría incluir acceso a los detalles de su tarjeta de crédito, ver detalles de contactos o cambiar contraseñas (Web Security, s. f.).

Clickjacking.

En este tipo de ataque, el usuario malicioso secuestra las pulsaciones de ratón dirigidas a un sitio visible por encima de los demás y las redirige a una página escondida por debajo. Esta técnica se usaría, por ejemplo, para presentar un sitio bancario legítimo pero capturar las credenciales de inicio de sesión en un *iframe* invisible controlado por el atacante. Alternativamente podría usarse para conseguir que el usuario pinchara sobre un botón en un sitio visible, pero al hacerlo realmente

estuviera sin advertirlo pinchando en otro botón completamente diferente. Como defensa, tu sitio puede protegerse de ser embebido en un *iframe* de otro sitio configurando las cabeceras HTTP apropiadamente.

Man-in-the-Middle (MitM).

Es un tipo de ataque destinado a interceptar, sin autorización, la comunicación entre dos dispositivos (*hosts*) conectados a una red. Este ataque le permite a un agente malintencionado manipular el tráfico interceptado de diferentes formas, ya sea para escuchar la comunicación y obtener información sensible, como credenciales de acceso, información financiera, etc., o para suplantar la identidad de alguna de las partes. Para que un ataque MitM funcione correctamente, el delincuente debe asegurarse que será el único punto de comunicación entre los dos dispositivos, es decir, el delincuente debe estar presente en la misma red que los hosts apuntados en el ataque para cambiar la tabla de enrutamiento para cada uno de ellos (*Qué es un ataque de Man-in-the-Middle y cómo funciona*, 2021).

Denegación de servicio (DoS).

La denegación de servicio se consigue inundando el sitio objetivo con peticiones de manera que, se interrumpa el acceso a los usuarios legítimos. Las peticiones pueden simplemente ser numerosas, o consumir individualmente gran cantidad de recursos. Las defensas contra DoS normalmente trabajan mediante la identificación y el bloqueo de tráfico "malo" permitiendo, sin embargo, que atraviesen los mensajes legítimos. Estas defensas se encuentran típicamente dentro o antes del servidor, no son parte de la aplicación web misma (*Seguridad de Sitios Web - Aprende sobre desarrollo web | MDN*, s. f.).

Phishing.

El *phishing* es una de las estafas más antiguas y mejor conocidas de internet. Se puede definir como un tipo de fraude en las telecomunicaciones que emplea trucos de ingeniería social para obtener datos privados de sus víctimas.

Un ataque de *phishing* tiene tres componentes:

- El ataque se realiza mediante comunicaciones electrónicas, como un correo electrónico o una llamada de teléfono.
- El atacante se hace pasar por una persona u organización de confianza.
- El objetivo es obtener información personal confidencial, como credenciales de inicio de sesión o números de tarjeta de crédito.

La mayor parte del *phishing* puede dar como resultado el robo de identidades o de dinero, y también es una técnica eficaz para el espionaje industrial y el robo de datos. Algunos *hackers* llegan incluso a crear perfiles falsos en redes sociales, invierten un tiempo en desarrollar una relación con las posibles víctimas y esperan a que exista confianza para hacer saltar la trampa. No solo hay daños financieros, en estos casos también se produce una pérdida de confianza (*Guía esencial del phishing*, s. f.).

1.2.2 Influencia de la seguridad en el posicionamiento web.

La inexistencia de medidas de seguridad en una web puede desembocar en consecuencias negativas para el SEO. Las principales quedan definidas a continuación:

- Desindexación de la web: En caso de detectarse que un sitio web ha sido infectado por algún tipo de malware, ingeniería social o similar puede ocurrir que el mismo sea desindexado completamente de los buscadores, ocasionando que el sitio pierda en gran medida el trafico orgánico, pudiendo llegar a producir consecuencias devastadoras para el mismo.
- Etiquetado del *Snippet* en los resultados: En el caso del motor de búsqueda de Google, al detectar que la seguridad de un sistema web se ha visto comprometida, añade un texto de advertencia para los usuarios en el *Snippet* de la web que muestra en las páginas de resultados. Aunque no impide el acceso a la web, la presencia de este texto ejercerá un efecto disuasorio en los usuarios, que preferirán acceder a otro portal antes que a uno infectado.
- Percepción de los usuarios: Aunque de forma indirecta, si la imagen de un sistema se ve dañada por el *hackeo*, repercutirá negativamente en el SEO.

La confianza en la empresa se vería mermada y disminuirían las búsquedas de marca.

- Imposibilidad de rastreo: Los ataques a una web, especialmente los de fuerza bruta o DDoS, pueden perjudicar al rendimiento e incluso dejar inactivo al servidor web. Esto implica que si en ese momento el *bot* de algún motor de búsqueda intenta rastrear un sitio web, no será capaz de encontrarlo (Osan, 2022).

1.3 Estudio de sistemas homólogos.

En este epígrafe se realiza un análisis de los principales sistemas homólogos con el objetivo de determinar cuáles son las variables de mayor peso relacionadas con la seguridad que dichos sistemas tienen en cuenta. Se describe primeramente cada sistema homólogo y se concluye con la extracción de las variables por cada uno de los sistemas.

1.3.1 Security Headers.

Security Headers fue creado por Scott Helme, investigador de seguridad y fundador de una empresa con sede en el Reino Unido.

El sitio evalúa los encabezados HTTP dando una calificación según la presencia de determinados parámetros y el grado de completitud de los mismos. Los encabezados de respuesta HTTP que analiza este sitio brindan altos niveles de protección y es importante que los sitios los implementen. Proporciona un mecanismo fácil para evaluarlos y más información sobre cómo implementar los encabezados que faltan, impulsando el uso de encabezados basados en la seguridad en la Web. El sitio está escrito en PHP utilizando el *framework CodeIgniter MVC* y funciona con *DigitalOcean Droplets* (Helme, s. f.).

1.3.2 Mozilla Observatory.

Mozilla Observatory es un conjunto de herramientas para analizar sitios web e informar si está utilizando los muchos métodos disponibles para protegerlo.

Se divide en tres proyectos:

- http-observatorio - escáner/graduador.

- http-observatory-cli - interfaz de línea de comandos.
- http-observatory-website - interfaz web.

La evaluación de TLS se basa en escáneres externos, como el Observatorio TLS de Mozilla.

Todos los sitios web comienzan con una puntuación de referencia de 100 y reciben penalizaciones o bonificaciones a partir de ahí. La puntuación mínima es 0, pero no hay una puntuación máxima. Los puntos de bonificación solo se otorgan si la puntuación del sitio sin ellos es 90 (A) o superior. Actualmente, la puntuación más alta posible en el Observatorio HTTP es 135.

Aunque tanto los rangos de calificación de letras como los modificadores son esencialmente arbitrarios, se basan en los comentarios de los profesionales de la industria sobre la importancia de aprobar o reprobar una prueba determinada (*Mozilla Observatory*, s. f.).

1.3.3 SSL Server Test.

Es un servicio en línea gratuito que realiza un análisis profundo de la configuración de cualquier servidor web SSL en la Internet pública. Teniendo en cuenta que la información que es enviada se utiliza únicamente para proporcionar el servicio.

Proyectos:

- API de SSL Labs: Las API de SSL Labs exponen la funcionalidad completa de prueba del servidor SSL/TLS de manera programática, lo que permite una evaluación programada y masiva.
- Mejores prácticas de implementación de SSL/TLS: El documento de mejores prácticas de implementación de SSL/TLS proporciona instrucciones claras y concisas para ayudar a los administradores y programadores con exceso de trabajo a dedicar el mínimo tiempo posible para implementar un sitio o una aplicación web seguros. La atención se centra en los consejos que son prácticos y fáciles de entender.
- Prueba de servidor SSL: La prueba del servidor SSL es un servicio en línea que le permite inspeccionar la configuración de cualquier servidor web SSL público.

- Prueba de cliente SSL: La prueba de cliente SSL muestra las capacidades SSL/TLS de su navegador.
- Pulso SSL: SSL Pulse es un panel continuo y global para monitorear la calidad del soporte SSL / TLS a lo largo del tiempo en 150,000 sitios web habilitados para SSL y TLS.
- Guía de clasificación de servidores SSL: SSL *Server Rating Guide* tiene como objetivo establecer una metodología de evaluación sencilla, que permita a los administradores evaluar la configuración del servidor SSL con confianza sin necesidad de convertirse en expertos en SSL.
- Capacidades del agente de usuario: Nuestra base de datos de agentes de usuario y sus capacidades SSL/TLS, que cubre una amplia gama de dispositivos, navegadores y herramientas populares.
- HTTP *client fingerprints* mediante análisis de protocolo de enlace SSL: Diferentes programas (que hacen uso de SSL) a menudo usan diferentes conjuntos de cifrado. Al observar la lista de conjuntos de cifrado admitidos, a menudo se puede adivinar la marca del cliente SSL del otro lado.
- Modelo de amenazas SSL: Un modelo de amenazas que cubre el ecosistema de seguridad SSL, que consta de SSL, TLS y PKI.

1.3.4 Resultados del estudio.

La tabla a continuación expone la relación entre las variables tomadas como propuesta de solución y cada uno de los sistemas homólogos de los cuales se extraen.

Tabla 1: Relación variable-sistema.

Variable / Sistema	Security Headers	Mozilla Observatory	SSL Server Test
Content Security Policy	X	X	
HTTP Strict Transport Security	X	X	X
Seguridad en las Cookies	X	X	
Referrer Policy	X	X	

Subresource Integrity		X	
Resource Loading		X	

Tras el análisis de cada uno de los anteriores sistemas homólogos se arriba a la conclusión del desarrollo de un grupo de variables relacionadas con la seguridad, las cuales mediante la unificación de los propios sistemas antes descritos le aportaran un alto nivel de completitud y profesionalidad al Monitor de Sitios Web cubano tras su integración al módulo del Evaluador.

1.4 Metodología, herramientas y técnicas utilizadas para el desarrollo de la solución.

Las metodologías, técnicas y herramientas expuestas en este epígrafe han sido seleccionadas debido a que la documentación y desarrollo del contenido se encuentre acorde con los desarrollos previos y futuros dentro del proyecto Monitor de Sitios Web cubano. De esta forma quedan resueltos posibles problemas de compatibilidad, asociación y colaboración con el resto de desarrollos en el ecosistema.

1.4.1 Metodología XP.

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos.

Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo.

Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes (Calvo 2018).

Características:

- Se considera al equipo de proyecto como el principal factor de éxito del proyecto.
- Software que funciona por encima de una buena documentación.

- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.
- Rápida respuesta a cambios.

Roles:

- **Cliente:** Responsable de definir y conducir el proyecto, así como sus objetivos.
- **Programadores:** Estiman tiempos de desarrollo de cada actividad y programan el proyecto.
- **Tester:** Encargado de Pruebas.
- **Tracker:** Encargado de Seguimiento.
- **Coach:** Entrenador. Su papel es guiar y orientar al equipo.
- **Big Boss:** Gestor del proyecto, gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado.

Se seleccionó la metodología XP ya que el macroproyecto Monitor de Sitios Web cubano se desarrolla utilizando dicha metodología de desarrollo de *software*, además del nivel de compatibilidad de sus principales características con las del proyecto en cuestión.

1.4.2 Framework y lenguaje de programación.

Spring mediante Spring Boot.

Spring Boot es una herramienta que nace con la finalidad de simplificar aún más el desarrollo de aplicaciones basadas en el ya popular *framework Spring Core*. *Spring Boot* busca que el desarrollador solo si centre en el desarrollo de la solución, olvidándose por completo de la compleja configuración que actualmente tiene *Spring Core* para poder funcionar.

Spring Boot centra su éxito en las siguientes características que lo hacen extremadamente fácil de utilizar:

- **Configuración:** *Spring Boot* cuenta con un complejo módulo que autoconfigura todos los aspectos de nuestra aplicación para poder simplemente ejecutar la aplicación, sin tener que definir absolutamente nada.

- **Resolución de dependencias:** Con *Spring Boot* solo hay que determinar qué tipo de proyecto estaremos utilizando y él se encarga de resolver todas las librerías/dependencias para que la aplicación funcione.
- **Despliegue:** *Spring Boot* se puede ejecutar como una aplicación *Stand-alone*, pero también es posible ejecutar aplicaciones web, ya que es posible desplegar las aplicaciones mediante un servidor web integrado, como es el caso de *Tomcat, Jetty* o *Undertow*.
- **Métricas:** Por defecto, *Spring Boot* cuenta con servicios que permite consultar el estado de salud de la aplicación, permitiendo saber si la aplicación está prendida o apagada, memoria utilizada y disponible, número y detalle de los *Bean's* creado por la aplicación, controles para el prendido y apagado, etc.
- **Extensible:** *Spring Boot* permite la creación de complementos, los cuales ayudan a que la comunidad de software libre cree nuevos módulos que faciliten aún más el desarrollo.

Java.

Es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Para la implementación del código de la aplicación se utilizó Java como lenguaje de programación. Un elemento también importante para la selección de este lenguaje se debe a que la herramienta sobre la cual se implementará la propuesta de solución está desarrollada con el mismo lenguaje de programación (Ramírez 2013).

A continuación, se muestran una serie de características donde se describe de forma más detallada las propiedades y funcionalidades que brinda este lenguaje de programación:

- **Orientado a objetos:** implementa la tecnología de C++ y soporta las tres características del paradigma orientado a objetos. Encapsulamiento: Implementa información oculta. Polimorfismo: El mismo mensaje se envía a diferentes objetos, resultando en comportamientos que dependen de la

naturaleza del objeto que recibió el mensaje. Herencia: Puede definir nuevas clases y comportamientos (métodos) basados en clases existentes.

- **Distribuido:** presenta extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Por si sólo no es distribuido, pero proporciona herramientas para que nuestros programas puedan serlo.
- **Simple:** Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. Elimina muchas de las características de otros lenguajes como C++, para mantener reducida la especificación del lenguaje.
- **Robusto:** Realiza verificaciones en busca de problemas, tanto en tiempo de compilación, como de ejecución. La comprobación de tipos ayuda a detectar errores. Obliga a la declaración explícita de los métodos.
- **Seguro:** La seguridad tiene las facetas: Se eliminan características como los apuntadores y el casting implícito para prevenir el acceso ilegal a la memoria El código Java pasa por muchas verificaciones antes de ser ejecutado en una máquina mediante el *classloade*.

1.4.3 Entorno de desarrollo.

Apache Netbeans.

Apache NetBeans es un proyecto Apache de nivel superior dedicado a proporcionar productos de desarrollo de software sólidos (*Apache NetBeans IDE y Apache NetBeans Platform*) que abordan las necesidades de los desarrolladores, usuarios y empresas que confían en *NetBeans* como base para sus productos; en particular, para permitirles desarrollar estos productos de manera rápida, eficiente y sencilla aprovechando las fortalezas de la plataforma Java y otros estándares relevantes de la industria.

Los dos productos básicos, *Apache NetBeans IDE y Apache NetBeans Platform*, son gratuitos para uso comercial y no comercial, bajo la licencia de *Apache*. El código fuente de ambos está disponible para que cualquiera pueda reutilizarlo como mejor le parezca, dentro de los términos de uso.

Con más de 18 millones de descargas de *NetBeans IDE* hasta la fecha y más de 800 000 desarrolladores participantes, el proyecto *NetBeans* prospera y sigue creciendo gracias a las personas y empresas asociadas.

La mayoría de los desarrolladores reconocen el IDE de *NetBeans* como el IDE gratuito original de Java. El IDE de *Apache NetBeans* brinda soporte para varios lenguajes (Java, PHP, JavaFX, JavaScript, etc.) y *frameworks*, y pronto se incorporarán más (C/C++).

La plataforma *Apache NetBeans* proporciona una arquitectura de aplicaciones confiable y flexible. Su aplicación no tiene que parecerse en nada a un IDE. La plataforma *NetBeans* le brinda una arquitectura probada en el tiempo de forma gratuita. Una arquitectura que fomenta prácticas de desarrollo sostenible. Debido a que la arquitectura de la plataforma *NetBeans* es modular, es fácil crear aplicaciones robustas y extensibles (*About Apache NetBeans*, s. f.).

1.4.4 Herramienta CASE.

Visual Paradigm.

Visual Paradigm es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de *software*: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia de usuario mediante el diseño de un gran número de artefactos de ingeniería de *software*. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de *software* y la modelación de procesos del negocio.

1.4.5 Control de versiones.

Gitlab.

Gitlab es un servicio web de control de versiones y desarrollo de software colaborativo basado en *Git*. Además de gestor de repositorios, el servicio ofrece también alojamiento de wikis y un sistema de seguimiento de errores, todo ello publicado bajo una licencia de código abierto (Torrado 2017).

Comenzó en 2011 como un proyecto de código abierto para ayudar a un equipo de programadores a colaborar, llegando a convertirse en una plataforma utilizada por millones de personas para entregar *software* de manera más rápida y eficiente, al tiempo que fortalecen la seguridad y el cumplimiento. Se centra en el trabajo remoto, el código abierto, *DevOps* y la iteración.

1.4.6 Gestor de bases de datos.

MySQL.

MySQL es una de las bases de datos más populares en el mundo, especialmente para desarrollos web. Fue desarrollada originalmente en los lenguajes C y C++ (Ramírez Navia 2018).

Como administrador de bases de datos relacional, *MySQL* almacena los datos en forma de tablas estructuradas, con campos, índices, llaves foráneas e integridad referencial en la actualización o borrado en cascada. Usa el lenguaje SQL para las diferentes transacciones de datos.

Se puede instalar prácticamente en cualquier sistema operativo: *Unix*, *Linux*, *Windows*, *MacOS*; solo o acompañado de otros programas del entorno de programación como el servidor web *Apache*, *PHP* y *Phpmyadmin*; en las distribuciones *WAMP, LAMP o XAMPP*.

MySQL se usa como base de datos en aplicaciones que requieren centralizar información, con datos que se pueden estructurar en tablas, con campos definidos, con índices y llaves que relacionan las tablas entre sí.

MySQL ha probado su desempeño en cantidades relativamente grandes de datos, manteniendo la operación de grandes plataformas, sin tener que incurrir en enormes sumas de dinero en licencias como ocurre con *SQL Server u Oracle*.

1.4.7 Gestor de colas de bloques de mensajes.

RabbitMQ.

RabbitMQ es un *software* de encolado de bloques de mensajes llamado *broker* de mensajería o gestor de colas. Dicho de forma simple, es un software donde se

pueden definir colas, las aplicaciones se pueden conectar a dichas colas y transferir/leer mensajes en ellas. Sus características principales son (Martín 2018):

- Garantía de entrega.
- Enrutamiento flexible.
- Clusterización.
- Federación.
- Alta disponibilidad.
- Tolerancia a fallos.

La arquitectura básica de una cola de mensajes es simple. Hay aplicaciones clientes, llamadas productores, que crean mensajes y los entregan al intermediario (la cola de mensajes). Otras aplicaciones, llamadas consumidores, se conectan a la cola y se suscriben a los mensajes que se procesarán. Un mensaje puede incluir cualquier tipo de información.

Conclusiones parciales.

La investigación realizada a cerca de los conceptos relacionados con la problemática permitió:

- Definir elementos relevantes para la gestión de la seguridad de sitios web y así estructurar el estudio de acuerdo con el objeto de estudio y el campo de acción definido.
- Definir los parámetros de mayor importancia que deberían ser evaluados por el Evaluador mediante la realización del análisis de las amenazas a la seguridad más comunes que afectan a los sitios web.
- Determinar cuáles son las variables de mayor importancia en cuanto a la evaluación de la seguridad de los sitios web mediante el estudio de los principales sistemas homólogos.
- Seleccionar XP como metodología de desarrollo y como herramienta CASE *Visual Paradigm* para el modelado, como lenguajes de programación: Java y *JavaScript*; utilizandar como *framework* y plataforma de desarrollo: *Spring Boot*; entorno de desarrollo: *Apache NetBeans*; para el control de versiones: *GitLab, MySql* para el almacenamiento de datos y *RabbitMQ* como *software*

de encolado de bloques de mensajes.

Capítulo 2: Aspectos relacionados con la implementación de variables de seguridad para el Monitor de Sitios Web cubano.

En el presente capítulo se expone la propuesta de solución partiendo de la problemática antes descrita, definiendo para ello el sistema en cuestión. Además, se especifican las historias de usuarios, el plan de iteraciones, el plan de entregas y las tarjetas Clase-Responsabilidad-Colaboración generadas como artefactos de la metodología. Se describe la arquitectura, así como los patrones de diseño utilizados. Se representa el diagrama de despliegue utilizado.

2.1 Propuesta de solución.

Tras definir como situación problemática la deficiencia del Monitor de Sitios Web cubano en cuanto a la evaluación de variables de seguridad de los sistemas web, se determinó como propuesta de solución el desarrollo e implementación de seis nuevas variables para la evaluación de la seguridad de los sistemas web. Para ello se realizó la recopilación de información mediante la entrevista realizada a diferentes especialistas en los sectores del desarrollo web y la seguridad informática, así como el análisis y estudio de los sistemas homólogos más destacados y las amenazas más comunes a la seguridad web.

A continuación se muestra una tabla que sintetiza los aspectos fundamentales de la implementación de dichas variables.

Tabla 2: Aspectos fundamentales de la implementación de las variables como propuesta de solución.

Variables	**Parámetros**	**Descripción**	**Posible Evaluación**		
Content Security Policy (CSP).	-Exista la Cabecera.	Si el sitio no implementa la cabecera se evalúa de 0(Mal), en caso de implementarla si contiene el atributo *unsafe-*	0	1	2

	-No contiene el atributo *unsafe-inline.*	*inline* se evalúa de 1(A mejorar) y en caso de que la implemente y no contenga dicho atributo se evalúa de 2(Bien).			
HTTP Strict Transport Security (HSTS).	-Exista la Cabecera. -Contiene el atributo *preload.*	Si el sitio no implementa la cabecera se evalúa de 0(Mal), en caso de implementarla si no contiene el atributo *preload* se evalúa de 1(A mejorar) y en caso de que la implemente y contenga dicho atributo se evalúa de 2(Bien).	0	1	2
Seguridad Cookies.	-Exista la Cabecera. -Contiene los atributos *httponly y secure.*	Si el sitio no implementa la cabecera o si la implementa y contiene ambos atributos se evalúa de 2(Bien), en caso de implementarla si no contiene uno de los atributos *httponly* o *secure* se evalúa de 1(A mejorar) y en caso de que la implemente y no contenga ninguno de los atributos se evalúa de 0(Mal).	0	1	2
Referrer Policy.	-Exista la Cabecera. -Contiene alguno de los atributos *no-referrer,* same-	Si el sitio no implementa la cabecera se evalúa de 0(Mal), en caso de implementarla si no contiene ninguno de los atributos *referrer,* same-origin, strict-origin o strict-	0	1	2

	origin, strict-origin o strict-origin-when-cross-origin.	origin-when-cross-origin se evalúa de 1(A mejorar) y en caso de que la implemente y contenga alguno de dichos atributos se evalúa de 2(Bien).			
Subresource Integrity.	-Atributo *integrity* de todos los javascript externos.	Si todos los javascrpt externos del sitio contienen el atributo *integrity* se evalúa de 1(Bien) en caso contrario se evalúa de 0(Mal).	0	1	
Resource Loading.	-Atributo *src* de todos los javascript externos(cargados mediante https).	Si todos los javascrpt externos del sitio son cargados mediante el protocolo https se evalúa de 1(Bien) en caso contrario se evalúa de 0(Mal).	0	1	

2.1.1 Descripción de las variables a implementar.

En el siguiente epígrafe se describe de forma más detallada cada uno de los aspectos relacionados con las variables a implementar como parte de la propuesta de solución definida.

Content Security Policy (CSP).

La política de seguridad de contenido (CSP) es un encabezado HTTP que permite a los operadores del sitio un control detallado sobre desde dónde se pueden cargar los recursos en su sitio. El uso de este encabezado es el mejor método para evitar vulnerabilidades de secuencias de comandos entre sitios (XSS). Debido a la dificultad de adaptar CSP a sitios web existentes, CSP es obligatorio para todos los

sitios web nuevos y se recomienda enfáticamente para todos los sitios existentes de alto riesgo (Web Security, s. f.).

El beneficio principal de CSP proviene de deshabilitar el uso de *JavaScript* en línea no seguro. El *JavaScript* en línea, ya sea reflejado o almacenado, significa que las entradas de usuario con escape incorrecto pueden generar código que el navegador web interpreta como *JavaScript*. Al usar CSP para deshabilitar *JavaScript* en línea, puede eliminar de manera efectiva casi todos los ataques XSS contra su sitio.

Teniendo en cuenta que deshabilitar *JavaScript* en línea significa que todo *JavaScript* debe cargarse desde las etiquetas <script> src, los controladores de eventos, como *onclick*, que se usan directamente en una etiqueta, no funcionarán, al igual que *JavaScript* dentro de las etiquetas <script> pero no se carga a través de src. Además, las hojas de estilo en línea que usan etiquetas <style> o el atributo de estilo tampoco se cargarán. Como tal, se debe tener cuidado al diseñar sitios para que CSP sea más fácil de implementar (Web Security, s. f.).

Aspectos de la implementación:

- Apuntar a default-src https: es un excelente primer objetivo, ya que deshabilita el código en línea y requiere HTTP.
- Para los sitios web existentes con bases de código grandes que requerirían demasiado trabajo para deshabilitar los scripts en línea, default-src https: *'unsafe-inline'* sigue siendo útil, ya que evita que los recursos se carguen accidentalmente a través de HTTP. Sin embargo, no proporciona ninguna protección XSS.
- Se recomienda comenzar con una política razonablemente bloqueada como *default-src 'none'; img-src 'onething'; script-src 'auto'; style-src 'self'* y luego agregue las fuentes como se reveló durante la prueba.
- En lugar del encabezado HTTP preferido, las páginas pueden incluir una etiqueta <meta http-equiv="Content-Security-Policy" content="…">. Si lo hacen, debería ser la primera etiqueta <meta> que aparece dentro de <head>.

- Se debe tener cuidado con los datos: URI, ya que no son seguros dentro de *script-src y object-src* (o se heredan de *default-src*).
- De manera similar, el uso de *script-src 'self'* puede no ser seguro para sitios con puntos finales *JSONP*. Estos sitios deben usar un *script-src* que incluya la ruta a su(s) carpeta(s) fuente de *JavaScript*.
- A menos que los sitios necesiten la capacidad de ejecutar complementos como *Flash* o *Silverlight*, deben deshabilitar su ejecución con *object-src 'none'*.
- Lo ideal es que los sitios usen la directiva *report-uri*, que *POSTs JSON* informa sobre las violaciones de CSP que ocurren. Esto permite detectar y reparar rápidamente las infracciones de CSP.
- Antes de la implementación, se recomienda usar el encabezado *HTTP Content-Security-Policy-Report-Only* para ver si se habría producido alguna infracción con esa política.

HTTP Strict Transport Security.

HTTP Strict Transport Security (HSTS) es un encabezado HTTP que notifica a los agentes de usuario que solo se conecten a un sitio determinado a través de HTTPS, incluso si el esquema elegido fue HTTP. Los navegadores que han configurado HSTS para un sitio determinado actualizarán de forma transparente todas las solicitudes a HTTPS. HSTS también le dice al navegador que trate los errores relacionados con TLS y certificados de manera más estricta al deshabilitar la capacidad de los usuarios para omitir la página de error.

El encabezado consta de un parámetro obligatorio (*max-age*) y dos parámetros opcionales (*includeSubDomains y preload*), separados por punto y coma (*Web Security*, s. f.).

Directivas:

- *max-age*: Cuánto tiempo los agentes de usuario redireccionarán a HTTPS, en segundos.
- *includeSubDomains*: Si los agentes de usuario deben actualizar las solicitudes en los subdominios.

- *Preload*: Si el sitio debe incluirse en la lista de precarga de HSTS.
- *max-age*: Debe establecerse en un mínimo de seis meses (15768000), pero se recomiendan períodos más largos, como dos años (63072000). Tenga en cuenta que una vez que se establece este valor, el sitio debe continuar admitiendo HTTPS hasta que se alcance el tiempo de vencimiento.
- *IncludeSubDomains*: Notifica al navegador que todos los subdominios del origen actual también deben actualizarse a través de HSTS. Por ejemplo, configurar *includeSubDomains* en dominio.mozilla.com también lo configurará en *host1.domain.mozilla.com* y *host2.domain.mozilla.com*. Se necesita mucho cuidado al configurar el indicador *includeSubDomains*, ya que podría deshabilitar sitios en subdominios que aún no tienen habilitado HTTPS.
- *Preload*: Permite que el sitio web se incluya en la lista de precarga de HSTS, una vez enviado. Como resultado, los navegadores web realizarán actualizaciones HTTPS en el sitio sin tener que recibir el encabezado HSTS inicial. Esto evita ataques de degradación en el primer uso y se recomienda para todos los sitios web de alto riesgo. Tenga en cuenta que estar incluido en la lista de precarga de HSTS requiere que también se configure *includeSubDomains*.

HttpOnly.

Las *cookies* que no requieren acceso desde *JavaScript* deben configurarse con la bandera *HttpOnly*. Cuando se usa el indicador *HttpOnly*, *JavaScript* no podrá leer la cookie en caso de explotación de XSS. También se analiza cómo se podría usar la combinación del método *HTTP TRACE* y *XSS* para omitir el indicador *HttpOnly*: esta combinación es un ataque de rastreo entre sitios. (*Web Security*, s. f.).

Secure.

Todas las *cookies* deben configurarse con el indicador *Secure*, lo que indica que solo deben enviarse a través de HTTPS. Estas banderas se pueden usar para hacer que las *cookies* sean más seguras. Cuando se usa un indicador seguro, la *cookie* solo se enviará a través de HTTPS, que es HTTP sobre SSL/TLS. Cuando este es

el caso, el atacante que espía el canal de comunicación del navegador al servidor no podrá leer la *cookie* (HTTPS proporciona autenticación, integridad de datos y confidencialidad) (*Web Security*, s. f.).

Subresource Integrity.

Subresource Integrity es un estándar W3C reciente que protege contra atacantes que modifican el contenido de las bibliotecas de *JavaScript* alojadas en redes de entrega de contenido (CDN) para crear vulnerabilidades en todos los sitios web que hacen uso de esa biblioteca alojada.

Por ejemplo, el código *JavaScript* en *jquery.org* que se carga desde *mozilla.org* tiene acceso a todo el contenido de *mozilla.org*. Si este recurso fue atacado con éxito, podría modificar los enlaces de descarga, desfigurar el sitio, robar credenciales, provocar ataques de denegación de servicio y más.

Subresource Integrity bloquea un recurso de *JavaScript* externo a su contenido conocido en un momento específico. Si el archivo se modifica en algún momento posterior, los navegadores web compatibles se negarán a cargarlo. Como tal, el uso de *Subresource Integrity* es obligatorio para todos los recursos de *JavaScript* externos cargados desde fuentes no alojadas en sistemas controlados por Mozilla.

Se debe tener en cuenta que las CDN deben ser compatibles con el estándar de uso compartido de recursos de origen cruzado (CORS) configurando el encabezado Access-Control-Allow-Origin (Web Security, s. f.).

Referrer Policy.

Cuando un usuario navega a un sitio a través de un hipervínculo o un sitio web carga un recurso externo, los navegadores informan al sitio de destino del origen de las solicitudes mediante el uso del encabezado HTTP Referer (sic). Aunque esto puede ser útil para una variedad de propósitos, también puede poner en riesgo la privacidad de los usuarios. La política de referencia HTTP permite que los sitios tengan un control detallado sobre cómo y cuándo los navegadores transmiten el encabezado de referencia HTTP (*Web Security*, s. f.).

En funcionamiento normal, si una página en *https://example.com/page.html* contiene <img src="https://not.example.com/image.jpg">, el navegador enviará una solicitud como esta:

GET/imagen.jpg HTTP/1.1
Host: not. example.com
Referer: https://example.com/page.html

Además de los riesgos para la privacidad que esto implica, el navegador también puede transmitir URL de uso interno que puede no haber tenido la intención de revelar. Si como operador del sitio, se limita la exposición de esta información, se puede usar la política de referencia HTTP para eliminar el encabezado de referencia o reducir la cantidad de información que contiene.

Directivas:

- no-referrer: Nunca envíe el encabezado Referer.
- same-origin: Enviar referente, pero solo en solicitudes al mismo origen.
- strict-origin: Envía la referencia a todos los orígenes, pero solo la URL sin la ruta (por ejemplo, https://example.com/)
- strict-origin-when-cross-origin: Envía la referencia completa en el mismo origen, URL sin la ruta en origen extranjero.

Resource Loading.

Todos los recursos, ya sea en el mismo origen o no, deben cargarse a través de canales seguros. Los navegadores bloquearán los sitios web seguros (HTTPS) que intenten cargar recursos activos como *JavaScript* de forma no segura. Como resultado, los usuarios experimentarán interfaces de usuario degradadas y advertencias de "contenido mixto". Los intentos de cargar contenido pasivo (como imágenes) de manera insegura, aunque son menos riesgosos, aún conducirán a interfaces de usuario degradadas y pueden permitir que los atacantes activos desfiguren sitios web o *phishing* a los usuarios.

A pesar de que los navegadores modernos hacen evidente que los sitios web están cargando recursos de manera insegura, estos errores aún ocurren con una frecuencia significativa. Para evitar que esto ocurra, los desarrolladores deben

verificar que todos los recursos estén cargados de forma segura antes de la implementación (*Web Security*, s. f.).

2.2 Historias de usuario.

Una historia de usuario es una explicación informal de una función de *software*, escrita desde la perspectiva del usuario final. Estas historias deben escribirse utilizando un lenguaje no técnico para brindar contexto al equipo de desarrollo (Asana, s. f.). Se definen las siguientes historias de usuario como las necesarias para la implementación de la propuesta de solución.

Tabla 3: HU_1.

Historia de Usuario	
Número: HU_1	**Usuario:** Administrador
Nombre de la historia: Identificar la cabecera CSP	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 1**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si un sitio implementa el encabezado CSP, si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	
Observaciones: En caso de que no se identifique el uso de la cabecera CSP, no se evalúa el otro indicador relacionado con esta variable.	

Tabla 4: HU_2.

Historia de Usuario	
Número: HU_2	**Usuario:** Administrador

Nombre de la historia: Validar la cabecera CSP	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada:** 1
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si en el encabezado CSP se encuentra presente el parámetro "*unsafe*", si lo utiliza se le otorga una evaluación de 0, sino se evalúa de 1 a ese indicador.	

El resto de las historias de usuario (HU_3 – HU_11) se podrán consultar en los anexos.

2.3 Plan de iteraciones.

Se genera una planificación en la que los desarrolladores junto al cliente delimitan los tiempos correctos para la implementación de las historias de usuarios, se establece la prioridad y cuáles de estas serán implementadas en cada versión del programa.

Tabla 5: Plan de iteraciones.

Iteración	Duración (Semanas)	Historias de Usuario
1	4	1-4
2	3	5-7
3	4	8-11

2.4 Plan de entregas.

Tabla 6: Plan de entregas.

Entregable	Duración (semanas)	Fecha de entrega
Identificar la existencia de la cabecera CSP, la cabecera HSTS y validar dichas cabeceras.	4	1 de Abril 2023
Identificar la seguridad en las cookies y validar dicha cabecera así como los parámetros necesarios para la seguridad.	3	22 de abril 2023
Identificar la cabecera Referrer Policy, la procedencia de los recursos cargados y validar los parámetros correspondientes a dicha cabecera, así como el parámetro Subresource Integrity.	4	20 de mayo 2023

Tras el análisis del tiempo de diseño e implementación necesario para cada una de las variables definidas, quedó conformado el plan de iteraciones, exponiendo como fecha de inicio el día 4 de marzo de 2023 y como fecha de culminación tras la última entrega el día 20 de mayo de 2023.

2.5 Tarjetas CRC.

Las tarjetas CRC que sus siglas significan: Clase – Responsabilidad – Colaborador son una herramienta que consiste en realizar un diagrama en el cual suele ir incluido: el nombre de la clase, lo que hace y quienes están involucrados; es decir, su función básica es determinar las clases orientadas a objetos de gran importancia y necesidad sobre el desarrollo del software. Dichas tarjetas solo hacen parte del proceso generado en XP (Saldaña Giraldo & Espinosa Valencia, 2021).

Las tarjetas CRC están divididas en 3 partes:

- **Clase:** Representa una colección de objetos similares.

- **Responsabilidades:** Describen las funciones que debe realizar una clase, es aquello que la clase sabe o hace.
- **Colaboraciones:** Describen las demás clases con las que trabaja una clase en conjunto para llevar a cabo sus responsabilidades.

Tabla 7: Tarjeta CRC “hsts”.

Tarjeta CRC	
Clase: hsts	
Responsabilidades	**Colaboraciones**
Identificar la cabecera HSTS. Validar la cabecera HSTS.	HttpURLConnection.

Tabla 8: Tarjeta CRC “cookies”.

Tarjeta CRC	
Clase: cookies	
Responsabilidades	**Colaboraciones**
Identificar la existencia de cookies. Validar el parámetro HttpOnly. Validar el parámetro Secure.	HttpURLConnection.

Tabla 9: Tarjeta CRC “referrer_policy”.

Tarjeta CRC	
Clase: referrer_policy	
Responsabilidades	**Colaboraciones**

Identificar la cabecera referrer policy. Validar en la cabecera Referrer Policy el parámetro no-referrer-when-downgrade. Validar en la cabecera Referrer Policy los parámetros no-referrer, same-origin, strict-origin y strict-origin-when-cross-origin.	HttpURLConnection.

El resto de las tarjetas CRC se podrán consultar en los anexos.

2.6 Patrones de diseño.

Son valorados debido a que imponen reglas sobre la arquitectura y expresan esquemas para solucionar problemas de un mismo tipo que pueden presentarse durante el desarrollo de la aplicación. Constituyen la base para realizar la búsqueda de soluciones a problemas que se presentan en el desarrollo de *software* y otros marcos del diseño de interacción. Una solución es considerada un patrón de diseño cuando posee ciertas características entre las cuales se encuentran: su efectividad debe haberse comprobado resolviendo problemas similares en otras ocasiones, debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Pressman, 2002).

Para el diseño de la propuesta de solución se tuvieron en cuenta los patrones Generales de *Software* para Asignación de Responsabilidades (GRASP). Entre ellos se encuentran experto en información, bajo acoplamiento, creador, alta cohesión, controlador.

Experto en información.

Este patrón plantea que se debe asignar una responsabilidad al experto en información, en otras palabras, a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma, se conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le corresponden de acuerdo a la información que poseen, lo que da lugar a sistemas más robustos y

fáciles de mantener (Larman, 2003). Este patrón se encuentra representado en la clase *download.java*, la cual tiene la información que se descarga de un sitio web.

Bajo acoplamiento.

El patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto (Larman, 2003).

Alta cohesión.

Este patrón plantea que se debe asignar una responsabilidad de modo que la cohesión siga siendo alta (una clase tiene responsabilidades moderadas). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. El patrón de alta cohesión se evidencia en las *SecureProtocolWorker.java* y *SecureProtocolOutput.java*, donde estas se hacen responsables de verificar que el protocolo sea el correcto y de almacenar en un objeto esa evaluación.

Creador.

Este patrón plantea que se debe asignar a una clase X la responsabilidad de crear una instancia de una clase Y. La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Este patrón es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Este patrón se pone de manifiesto en la clase *SecureProtocolOutput.java*, ya que es la encargada de crear la estructura que almacenará la evaluación de esa variable.

2.7 Arquitectura del software.

Para el desarrollo de los procesos es necesario definir la arquitectura de software a utilizar que se puede definir como: la estructura de un sistema o bien la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos (Pressman, 2002).

La arquitectura representa la clave para comprender, organizar y comunicar un sistema, además, facilita la evolución de la solución. Es diseñada para satisfacer los

requerimientos funcionales y no funcionales establecidos por los usuarios, clientes y proveedores del sistema. Un *software* que no posee un correcto diseño arquitectónico puede funcionar de forma muy deficiente o simplemente no funcionar generando consecuencias para la organización que sirve. Para las empresas que dependen de los sistemas de información las arquitecturas de *software* son fundamentales para el logro de sus objetivos organizacionales, lo que incluye el poder evolucionar rápidamente según las condiciones altamente cambiantes de los mercados actuales (Villegas, 2015).

Para definir la arquitectura de los procesos que se van a desarrollar se debe tener presente los patrones asociados.

2.7.1 Patrones de arquitectura.

Ofrecen soluciones a problemas de determinada arquitectura de software. Facilitan una descripción de los elementos y el tipo de relación que poseen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico formula un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (Pressman, 2002). Algunos de estos patrones son: programación por capas, tres niveles, pipeline, invocación implícita, arquitectura dirigida por eventos, arquitectura orientada a servicios, modelo vista controlador, entre otros patrones. Para el desarrollo de los procesos para el Monitor de Sitios Web Cubanos se seleccionó la arquitectura de microservicios.

2.7.2 Arquitectura de microservicios.

Los microservicios son un tipo de arquitectura que sirve para diseñar aplicaciones. Lo que distingue a la arquitectura de microservicios de los enfoques tradicionales y monolíticos es la forma en que desglosa una aplicación en sus funciones principales. Cada función se denomina servicio y se puede diseñar e implementar de forma independiente. Esto permite que funcionen separados sin afectar a los demás. Las aplicaciones complejas se componen de procesos independientes

pequeños que se comunican entre sí mediante API que no utilizan el mismo idioma (Vargas, 2019).

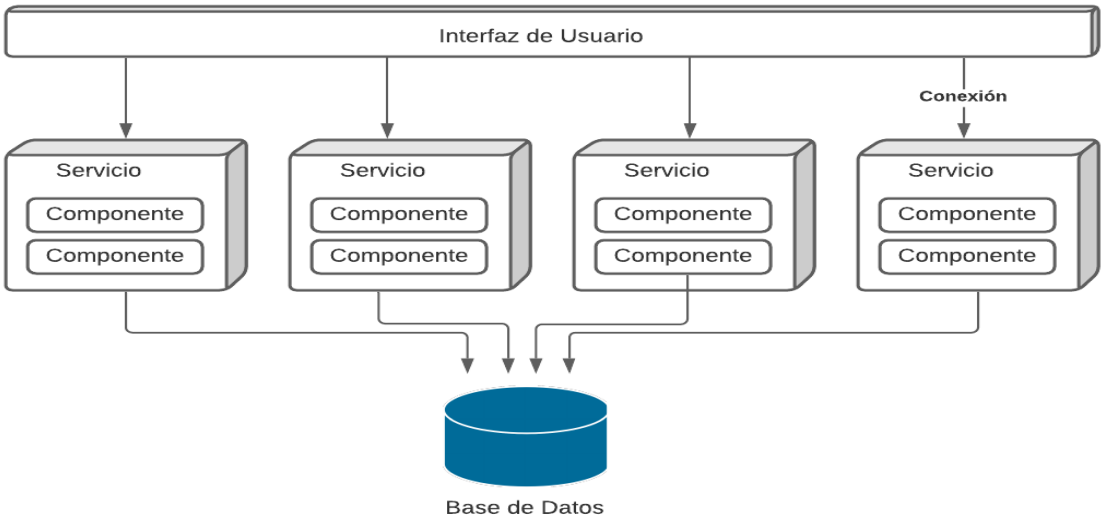


Figura 1: Arquitectura de microservicios. Fuente: Extraída de https://google.com/

2.8 Diagrama de despliegue.

Es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue o distribución de los artefactos del software en los destinos en que se propone el mismo. Define a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son los archivos ejecutables, bibliotecas, archivos, esquemas de base de datos y archivos de configuración. Cuando se hace mención de destino de despliegue no es más que un nodo el cual es un dispositivo de hardware o un entorno de ejecución de software (Sarmiento 2016).

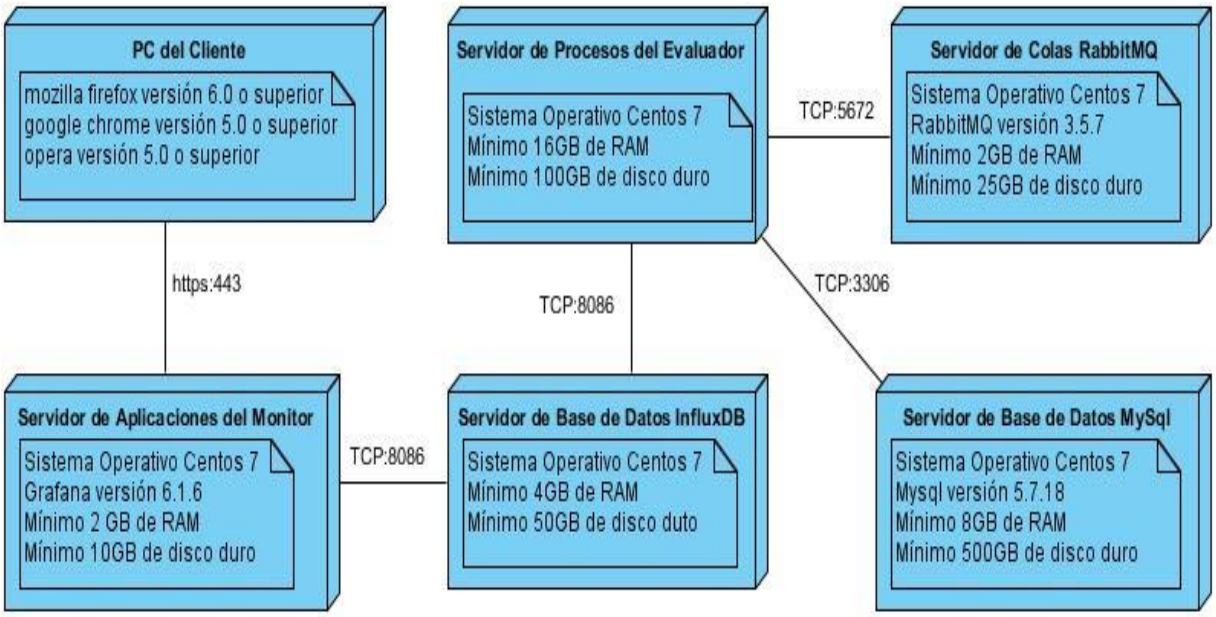


Figura 2: Diagrama de despliegue. Fuente: Dirección de Proyectos Especiales.

2.8.1 Nombre del procesador: descripción de la funcionalidad y capacidad del nodo.

Nodo: Elementos de procesamiento con al menos un procesador, memoria.

Ejemplos:

- **PC del Cliente:** Son todos los dispositivos electrónicos que puedan acceder a la aplicación del monitor para consultar la evaluación de su sitio web.
- **Servidor de Aplicaciones del Monitor:** Se encarga del procesamiento y control de la información que se encuentre en la aplicación del grafana.
- **Servidor de Base de Datos InfluxDB:** Se encarga de almacenar las evaluaciones de los sitios web que fueron evaluados por el evaluador, para que sean utilizadas por el servidor de aplicaciones del monitor.
- **Servidor de Base de Datos MySql:** Se encarga de almacenar las evaluaciones de los sitios web que fueron evaluador por el evaluador, para que sean utilizadas por la aplicación SeoWebMas.
- **Servidor de Procesos del Evaluador:** Se encarga de ejecutar todos los procesos de evaluación de las variables.

- **Servidor de Colas RabbitMQ:** Es un servidor que se encarga de definir colas, en el cual las aplicaciones se pueden conectar y transferir/leer mensajes entre ellas.

2.8.2 Nombre del tipo de conexión: Características físicas de la conexión.

Conectores: Expresa el tipo de conector o protocolo que relaciona dos elementos.

Ejemplos:

- **Https:443:** Protocolo que se utiliza para conectar el nodo PC del cliente y el servidor de aplicaciones del monitor.
- **TCP: 8086:** Protocolo que se utiliza para conectar los nodos servidor de aplicaciones del monitor y el servidor de procesos del evaluador con el servidor de base de datos InfluxDB.
- **TCP: 5672:** Protocolo que se utiliza para conectar el nodo servidor de procesos del evaluador con el servidor de colas RabbitMQ.
- **TCP: 3306:** Protocolo que se utiliza para conectar el nodo servidor de procesos del evaluador con el servidor de base de datos MySql.

Conclusiones parciales.

- Quedó planteada la propuesta de solución para el problema de investigación existente.
- Se identificaron un total de 11 historias de usuario, en las cuales se explicaron cada una de las funciones del software a desarrollar de manera detallada.
- Se desarrollaron el plan de iteraciones, el plan de entregas y las tarjetas CRC como artefactos de la propia metodología.
- Se establecieron los patrones de diseño GRASP con el objetivo de facilitar la reutilización del código en futuras versiones.
- Se realizó la identificación de la arquitectura para lograr una mayor organización de los elementos que conformarán la aplicación.
- Se expuso un diagrama de despliegue que permitió identificar las diferentes topologías de hardware sobre las cuales se ejecutará el sistema.

Capítulo 3: Resultados del desarrollo de las variables de seguridad para el Monitor de Sitios Web cubano.

En el presente capitulo se presentarán las tareas de ingeniería respectivas para cada historia de usuario, así como los diferentes tipos de pruebas realizadas y los criterios utilizados para su diseño y ejecución. Se analizarán los resultados obtenidos, exponiendo las posibles no conformidades y sus soluciones. Estas pruebas proporcionarán una evaluación exhaustiva del software y permitirán determinar su nivel de funcionalidad.

3.1 Tareas de ingeniería.

La descomposición de las Historias de Usuario en tareas de ingeniería implica describir las actividades que se llevarán a cabo para cada historia de usuario en particular. Estas tareas están estrechamente relacionadas con el trabajo del desarrollador y proporcionan una oportunidad para involucrarse directamente con el código. Cada tarea de ingeniería representa una actividad específica que debe realizarse, probarse y completarse como parte de la historia de usuario (MELÉNDEZ VALLADAREZ et al., 2016).

Tabla 10: Tarea de Ingeniería Implementar la funcionalidad que permita identificar la cabecera CSP.

Tarea de Ingeniería	
Número de la tarea: 1	**Nombre de la HU:** Identificar la cabecera CSP
Nombre de la tarea: Implementar la funcionalidad que permita identificar la cabecera CSP	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 4 de Marzo 2023	**Fecha fin:** 11 de Marzo 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar el método de la clase CSP que permite determinar si un sitio implementa el encabezado CSP, si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador y no mide el resto de indicadores de dicha variable.	

Tabla 11: Tarea de Ingeniería Implementar la funcionalidad validar la cabecera CSP.

Tarea de Ingeniería

Número de la tarea: 2	**Nombre de la HU:** Validar la cabecera CSP
Nombre de la tarea: Implementar la funcionalidad validar la cabecera CSP	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 11 de Marzo 2023	**Fecha fin:** 18 de Marzo 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar el método de la clase CSP que permite determinar si se encuentra presente el parámetro “unsafe”, si lo utiliza se le otorga una evaluación de 0, sino se evalúa de 1 a ese indicador que tributa a la evaluación final de la variable.	

Tabla 12: Tarea de Ingeniería Implementar la funcionalidad que permita Identificar la cabecera HSTS.

Tarea de Ingeniería	
Número de la tarea: 3	**Nombre de la HU**: Identificar la cabecera HSTS
Nombre de la tarea: Implementar la funcionalidad que permita identificar la cabecera HSTS	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 18 de Marzo 2023	**Fecha fin:** 25 de Marzo 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar la funcionalidad de la clase HSTS que permite conocer si un sitio implementa el encabezado HSTS, si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador y no evalúa el resto de los indicadores relacionados con dicha variable.	

Tabla 13: Tarea de Ingeniería Implementar la funcionalidad que permita Validar la cabecera HSTS.

Tarea de Ingeniería	
Número de la tarea: 4	**Nombre de la HU:** Validar la cabecera HSTS
Nombre de la tarea: Implementar la funcionalidad que permita validar la cabecera HSTS	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 25 de Marzo 2023	**Fecha fin:** 1 de Abril 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar la funcionalidad de la clase HSTS que permite conocer si en el encabezado HSTS se encuentra presente el parámetro “preload”, si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

El resto de las tareas de ingeniería se podrán consultar en los anexos.

3.2 Validación de la propuesta de solución.

Uno de los pilares de XP es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

Las pruebas del sistema tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema. Generalmente las pruebas del sistema son desarrolladas por los programadores para verificar que su sistema se comporta de la manera esperada, por lo que podrían encajar dentro de la definición de pruebas unitarias que propone XP (Gutiérrez et al., s. f.).

3.3 Pruebas funcionales.

Las pruebas funcionales son utilizadas para verificar si el desarrollo cumple con los requerimientos establecidos por el cliente. Estas pruebas se basan en historias de usuario y criterios de aceptación, los cuales permiten verificar de manera detallada la funcionalidad y el cumplimiento de los requerimientos desde su concepción. Una de las ventajas de este enfoque es que al seguir un orden verificable con las historias de usuario, los requerimientos son entregados de acuerdo a lo especificado (Acuña & López, 2022).

Para la realización de las pruebas funcionales se separó a cada variable implementada en un caso de prueba y se desarrolló una tabla por cada caso para describir el proceso. La celda de las tablas referente a la evaluación de cada caso contiene V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato.

Caso de Prueba 1: Proceso de evaluación de la variable Content Security Policy.

Descripción general: Prueba a la funcionalidad de evaluación de la variable Content Security Policy.

Tabla 2: Caso de prueba proceso de evaluación de la variable Content Security Policy.

Escenario	Descripción	Evaluación	Respuesta del sistema	Flujo central
EC 1.1 Realizar la evaluación al sitio web https://www.facebook.com/	El sistema realiza la evaluación de la variable Content Security Policy al sitio web **https://www.facebook.com/** de manera correcta, indicando si se encuentra presente la cabecera y no contiene el parámetro *unsafe.*	V	El sistema realiza la evaluación de la variable Content Security Policy de forma correcta,	El sistema manda a evaluar el sitio web **https://www.facebook.com/**
		Evaluación correcta.		

			indicando que se encuentra presente la cabecera aunque arroja como resultado mejorar la seguridad del sitio al encontrarse presente el parámetro *unsafe*.	

EC 1.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	El sistema realiza la evaluación de la variable Content Security Policy al sitio web **https://observatory.mozilla.org/** de manera correcta, indicando si se encuentra presente la cabecera y no contiene el parámetro *unsafe.*	V Evaluación correcta.	El sistema realiza la evaluación de la variable Content Security Policy de forma correcta, indicando que se encuentra presente la cabecera y arroja como resulta	**EC 1.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/**

			do una evaluación correcta al no encontrarse presente el parámetro *unsafe*.	
EC 1.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	El sistema realiza la evaluación de la variable Content Security Policy al sitio web **https://visuales.uclv.cu/** de manera correcta, indicando si se encuentra presente la cabecera y no contiene el parámetro *unsafe.*	V Evaluación correcta.	El sistema realiza la evaluación de la variable Content Security Policy de forma	El sistema manda a evaluar el sitio web **https://visuales.uclv.cu/**

			correcta, indicando que no se encuentra presente la cabecera arrojando como resultado una evaluación de mal.	

Descripción de las variables.

Tabla 3: Descripción de las variables del caso de prueba proceso de evaluación de la variable Content Security Policy.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tiene en la cabecera Content Security Policy.	Campo de texto.	No	Permite conocer si el sitio web tiene en la cabecera Content Security Policy.

2	Tiene válida la cabecera Content Security Policy.	Campo de texto.	No	Permite conocer si el sitio web tiene válida la cabecera Content Security Policy.

Caso de Prueba 2: Proceso de evaluación de la variable Strict Transport Security.

Descripción general: Prueba a la funcionalidad de evaluación de la variable Strict Transport Security.

Tabla 4: Caso de prueba proceso de evaluación de la variable Strict Transport Security.

Escenario	**Descripción**	**Evaluación**	**Respuesta del sistema**	**Flujo central**
EC 2.1 Realizar la evaluación al sitio web https://www.facebook.com/	El sistema realiza la evaluación de la variable Strict Transport Security al sitio web **https://www.facebook.com/** de manera correcta, indicando si se encuentra presente la cabecera y contiene el parámetro *preload.*	V Evaluación correcta.	El sistema realiza la evaluación de la variable Strict Transport Security de	El sistema manda a evaluar el sitio web **https://www.facebook.com/**

			forma correcta, indicando que se encuentra presente la cabecera y arroja como resultado una evaluación correcta al encontrarse presente el parámetro *preload.*	

EC 2.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	El sistema realiza la evaluación de la variable Strict Transport Security al sitio web **https://observatory.mozilla.org/** de manera correcta, indicando si se encuentra presente la cabecera y contiene el parámetro *preload.*	V Evaluación correcta.	El sistema realiza la evaluación de la variable Strict Transport Security de forma correcta, indicando que se encuentra presente la cabecera y arroja como resultado una	**EC 1.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/**

			evaluación correcta al encontrarse presente el parámetro *preload.*	
EC 2.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	El sistema realiza la evaluación de la variable Strict Transport Security al sitio web **https://visuales.uclv.cu/** de manera correcta, indicando si se encuentra presente la cabecera y contiene el parámetro *preload.*	V Evaluación correcta.	El sistema realiza la evaluación de la variable Strict Transport Security de forma correcta,	El sistema manda a evaluar el sitio web **https://visuales.uclv.cu/**

			indicando que no se encuentra presente la cabecera y arroja como resultado una evaluación de mal.	

Descripción de las variables.

Tabla 5: Descripción de las variables del caso de prueba proceso de evaluación de la variable Strict Transport Security.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tiene en la cabecera Strict Transport Security.	Campo de texto.	No	Permite conocer si el sitio web tiene en la cabecera Strict Transport Security.

2	Tiene válida la cabecera Strict Transport Security.	Campo de texto.	No	Permite conocer si el sitio web tiene válida la cabecera Strict Transport Security.

El resto de los casos de pruebas para las pruebas funcionales se podrán consultar en los anexos.

3.3.1 Resultados de las pruebas funcionales.

Al realizar las pruebas de funcionalidad se detectó como no conformidad que no existía estandarización con respecto al uso de mayúsculas al momento de declarar la cabecera set-cookies, por tanto ocurrió un fallo al no reconocer dicha cabecera en el sitio ***http://facebook.com/***, dando como resultado una evaluación negativa cuando realmente se encontraba presente dicha cabecera. Se corrigió el error y se ejecutó nuevamente la prueba que arrojo el resultado esperado. No se detectaron no conformidades en el resto de las iteraciones.

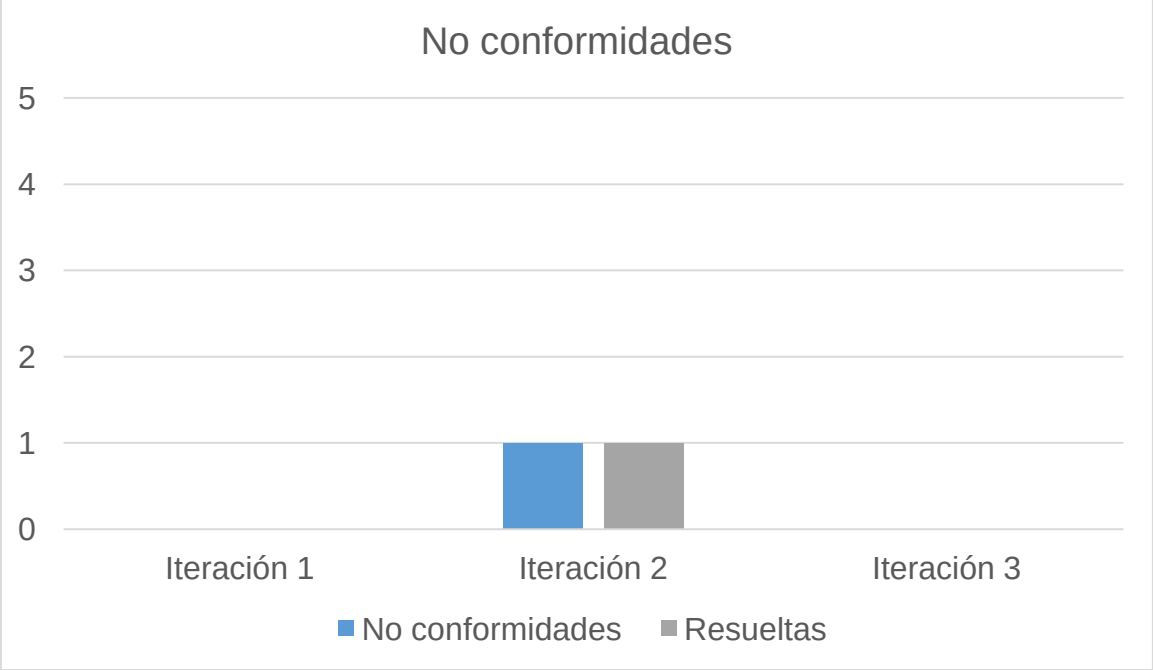


Figura 3: Gráfica de no conformidades. Elaboración: propia.

3.4 Pruebas unitarias.

Los programadores se encargan de llevar a cabo las pruebas del sistema con el fin de verificar la conformidad del sistema con las especificaciones. Estas pruebas se alinean con el concepto de pruebas unitarias dentro de la metodología XP (Gutiérrez et al., s. f.).

Las tablas a continuación describen el proceso de pruebas unitarias realizadas a las variables desarrolladas, las cuales se realizan de manera automatizada mediante el uso de la herramienta *JUnit*.

Tabla 6: Evaluación del método ContentSecurityPolicy de la clase csp.

Escenario	Respuesta del sistema	Respuesta esperada
EC 6.1 Realizar la evaluación al sitio web https://www.facebook.com	1	1
EC 6.2 Realizar la evaluación al sitio web https://observatory.mozzilla.org	2	2
EC 6.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	0	0

Tabla 7: Evaluación del método StrictTransportSecurity de la clase hsts.

Escenario	Respuesta del sistema	Respuesta esperada
EC 6.1 Realizar la evaluación al sitio web https://www.facebook.com	2	2
EC 6.2 Realizar la evaluación al sitio web https://observatory.mozzilla.org	1	1
EC 6.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	0	0

Tabla 8: Evaluación del método cookies de la clase cookies.

Escenario	Respuesta del sistema	Respuesta esperada
EC 6.1 Realizar la evaluación al sitio web https://www.facebook.com	2	2

EC 6.2 Realizar la evaluación al sitio web https://observatory.mozzilla.org	2	2
EC 6.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	2	2

El resto de las pruebas unitarias se podrán consultar en los anexos.

Consultar la Figura 5 sobre los resultados de las pruebas unitarias realizadas con JUnit en los anexos.

3.5 Pruebas de integración.

Las pruebas de integración tienen como objetivo evaluar la interacción entre los diversos componentes de un sistema, así como verificar la funcionalidad y la integración entre dos o más sistemas. Estas pruebas permiten tomar los componentes que han sido probados de forma individual y construir una estructura acorde al diseño propuesto (Gutiérrez et al., s. f.). En el caso específico de los microservicios desarrollados, se llevaron a cabo pruebas de integración dentro del macroproyecto "Monitor de Sitios Web Cubanos" para garantizar su correcto funcionamiento. Se realizaron procesos de evaluación en tres sitios web, y en cada caso, el sistema recibió y procesó correctamente los datos y evaluaciones de los sitios web, agregándolos a la base de datos. Esto evidencia que los microservicios están preparados para integrarse de manera exitosa al macroproyecto.

Tabla 9: Pruebas de integración.

Escenario	Descripción	Manual de Configuración	Valor Esperado	Respuesta de la Aplicación
Evaluar la cabecera Content Security Policy.	El sistema realiza la evaluación del indicador referente a la cabecera CSP y envía la información a la base de datos, para posteriormente mostrarla en la aplicación SeoWebMas.	El sitio web tiene que estar disponible antes de realizar la evaluación.	2 si es correcta, 1 si es a mejorar y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y mostrarlo en la aplicación SeoWebMas.
Evaluar la cabecera Http Strict Transport Security.	El sistema realiza la evaluación del indicador referente a la cabecera HSTS y envía la información a la base de datos, para posteriormente mostrarla en la	El sitio web tiene que estar disponible antes de realizar la evaluación.	2 si es correcta, 1 si es a mejorar y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y mostrarlo en la

	aplicación SeoWebMas.			aplicación SeoWebMas.
Evaluar la cabecera Cookies.	El sistema realiza la evaluación del indicador referente a la cabecera Cookies y envía la información a la base de datos, para posteriormente mostrarla en la aplicación SeoWebMas.	El sitio web tiene que estar disponible antes de realizar la evaluación.	2 si es correcta, 1 si es a mejorar y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y mostrarlo en la aplicación SeoWebMas.
Evaluar la cabecera Referrer Policy.	El sistema realiza la evaluación del indicador referente a la cabecera RP y envía la información a la base de datos, para posteriormente mostrarla en la	El sitio web tiene que estar disponible antes de realizar la evaluación.	2 si es correcta, 1 si es a mejorar y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y mostrarlo en la

	aplicación SeoWebMas.			aplicación SeoWebMas.
Evaluar la procedencia segura de los recursos cargados.	El sistema realiza la evaluación del indicador referente a la procedencia segura de los recursos cargados y envía la información a la base de datos, para posteriormente mostrarla en la aplicación SeoWebMas.	El sitio web tiene que estar disponible antes de realizar la evaluación.	1 si es correcta y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y mostrarlo en la aplicación SeoWebMas.
Evaluar el parámetro Integrity de los recursos cargados.	El sistema realiza la evaluación del indicador referente a la cabecera CSP y envía la información a la base de datos, para	El sitio web tiene que estar disponible antes de realizar la evaluación.	1 si es correcta y 0 si es incorrecta la evaluación.	El sistema verifica que el sitio esté disponible, para posteriormente evaluarlo, almacenarlo en la base de datos y

	posteriormente mostrarla en la aplicación SeoWebMas.			mostrarlo en la aplicación SeoWebMas.

Conclusiones parciales.

- Se establecieron de forma clara las funcionalidades a las cuales se le realizaron las pruebas tras la definición de las tareas de ingeniería.
- Se detectó una no conformidad a lo largo del proceso de pruebas funcionales, la cual fue corregida, permitiendo obtener el resultado esperado.
- Se realizaron las pruebas unitarias, arrojando un resultado excelente al no detectarse errores o fallas en el código de las variables implementadas.

Conclusiones Generales.

- El estudio teórico-metodológico y el desarrollo de los artefactos metodológicos permitieron desarrollar un total de seis nuevas variables aportándole un mayor grado de completitud y profesionalidad al Monitor de Sitios Web cubano.
- Las pruebas aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el sistema constituye una solución funcional.
- La validación del problema de investigación mediante la carta de aceptación emitida por el cliente (Dirección de Proyectos Especiales) demostró que las variables implementadas contribuyen un resultado satisfactorio para la solución de dicho problema.

Recomendaciones.

Se recomienda continuar con la implementación de nuevas variables tomando como referencia cada uno de los principales evaluadores web existentes y así perfeccionar el Monitor de Sitios Web cubano.

Referencias bibliográficas.

About Apache NetBeans. (s. f.). Recuperado 4 de abril de 2023, de https://netbeans.apache.org/about/index.html

Acuña, R. A. P., & López, Y. Q. (2022). *DISEÑAR DE LA METODOLOGÍA DE PRUEBAS FUNCIONALES PARA LOS DESARROLLOS EN FIDELITY MARKETING SAS*.

Arbeláez Gómez, M. C. (2014). Las tecnologías de la información y la comunicación (TIC) un instrumento para la investigación. *Investigaciones Andina*, *16*(29), 997-1000.

Asana. (s. f.). *¿Qué es la programación extrema (XP)? [2022]*. Asana. Recuperado 11 de mayo de 2023, de https://asana.com/es/resources/extreme-programming-xp

Botton, E. (2018). *Social Media, SEO and Google Ads in Digital Marketing Strategy: A Case Study on EcorNaturaSì*. http://dspace.unive.it/handle/10579/13829

Code Conventions for the Java Programming Language: 1. Introduction. (s. f.). Recuperado 16 de septiembre de 2023, de https://www.oracle.com/java/technologies/javase/codeconventions-introduction.html#16712

Gonzalez Garay, S., Sánchez Michel, G., & Ramírez Reyes, M. (2021). Análisis del posicionamiento web en portales web: Casos de estudio y buenas prácticas. *Revista Cubana de Ciencias Informáticas*, *15*(4), 125-140.

Guía esencial del phishing: Cómo funciona y cómo defenderse. (s. f.). Guía esencial del phishing: cómo funciona y cómo defenderse. Recuperado 3 de abril de 2023, de https://www.avast.com/es-es/c-phishing

Gutiérrez, J. J., Escalona, M. J., Mejías, M., & Torres, J. (s. f.). *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*.

Helme, S. (s. f.). *Analyse your HTTP response headers*. Recuperado 29 de marzo de 2023, de https://securityheaders.com/

MELÉNDEZ VALLADAREZ, S. M., ELIZABETH GAITAN, M., & NOEL PÉREZ REYES, N. (2016). *Metodología Ágil Programación Extrema XP*.

Molina Hernández, Y., Granda Dihigo, A., VelázquezCintra, A., Molina Hernández, Y., Granda Dihigo, A., & VelázquezCintra, A. (2020). Estrategia de desarrollo de requisitos no funcionales en aplicaciones para la salud. *Revista Cubana de Informática Médica*, *12*(1), 92-107.

Monitor RedCuba: Una web segura, visible y optimizada (+ Video) | Universidad de las Ciencias Informáticas. (s. f.). Recuperado 4 de abril de 2023, de https://www.uci.cu/universidad/noticias/monitor-redcuba-una-web-segura-visible-y-optimizada-video

Mozilla Observatory. (s. f.). Recuperado 29 de marzo de 2023, de https://observatory.mozilla.org/?

Niño Benitez, Y., & Silega Martínez, N. (2018). Requisitos de Seguridad para aplicaciones web. *Revista Cubana de Ciencias Informáticas*, *12*, 205-221.

Osan, G. (2022, enero 29). *SEO y Seguridad web, más allá del HTTPs | Latevaweb*. https://www.latevaweb.com/seo-y-seguridad-web

Qué es un ataque de Man-in-the-Middle y cómo funciona. (2021, diciembre 28). WeLiveSecurity. https://www.welivesecurity.com/la-es/2021/12/28/que-es-ataque-man-in-the-middle-como-funciona/

Saldaña Giraldo, A. D., & Espinosa Valencia, J. L. (2021). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN WEB RESPONSIVE ORIENTADO A LAS TAREAS ADMINISTRATIVAS PARA EL GIMNASIO JOSÉ SPORT GYM* [Programa

de Ingeniería de Sistemas y Telecomunicaciones. Pereira]. Universidad Católica de Pereira.

Seguridad de Sitios Web—Aprende sobre desarrollo web | MDN. (s. f.). Recuperado 23 de marzo de 2023, de https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Website_security

SEO - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN. (s. f.). Recuperado 20 de marzo de 2023, de https://developer.mozilla.org/es/docs/Glossary/SEO

Soriano, P. M. G. (2021, junio 25). A cuatro años del Monitor de sitios web en Cuba. *Cubaperiodistas*. https://www.cubaperiodistas.cu/2021/06/a-cuatro-anos-del-monitor-de-sitios-web-en-cuba/

Torres Pombert, A. (2003). El uso de los buscadores en Internet. *ACIMED*, *11*(3), 7-8.

Web Security. (s. f.). Recuperado 15 de marzo de 2023, de https://infosec.mozilla.org/guidelines/web_security#content-security-policy

Anexos.

Entrevista.

Personas entrevistadas:

1. Especialista "A" en Ciencias Informáticas Ramón Morales Alvarez del Centro de Innovación y Desarrollo de Internet.
2. M. Cs. Henry Raúl González Brito Subdirector del Centro de Telemática.
3. Especialista Dargel Veloz Morales del Departamento de Sistemas Digitales.

Guía de preguntas:

4. ¿Cuáles considera usted que son los puntos más significativos dentro de la seguridad de los sitios web?
5. ¿Cuántos parámetros influyen dentro de cada uno de los puntos antes descritos y cuáles son?
6. ¿Cómo se engloban cada uno de los puntos antes descritos junto con sus respectivos parámetros dentro de una variable a implementar?
7. ¿Algunas de las variables antes descritas tienen relación entre ellas?
8. ¿Cuáles de las variables definidas considera de mayor relevancia en cuanto al proceso de evaluación de los sitios web?

Guía de observaciones.

Nombre del observador: José Luis Domínguez Echevarría.

Fecha: 1 de marzo de 2023 Lugar: Dirección de Proyectos Especiales.

Hora de inicio: 0800 horas Hora final: 1600 horas

Propósito: Comprender el sistema de evaluación de variables por parte del Evaluador del Monitor de Sitios Web cubano y determinar carencias en cuanto a este proceso.

Tabla 21: Guia de Observaciones.

No.	ítem	Si	No	Observación

1	La arquitectura del sistema es basada en microservicios.	X		El sistema de evaluación del monitor presenta una arquitectura basada en microservicios, aportando desagregación.
2	Las variables se evalúan de forma cualitativa.		X	La evaluación de las variables se realiza de forma cuantitativa, dando 0 en caso de estar mal, 1 en caso de a mejorar y 2 en caso de estar bien.
3	Existe proporción en cuanto a la evaluación de los diferentes parámetros en los sitios web.		X	Existe una desproporción notable en cuanto a la evaluación de los diferentes parámetros en los sitios web, dejando mucho que desear con respecto a la evaluación de los parámetros de seguridad.
4	Influye de forma negativa la carencia de en cuanto a los parámetros de seguridad en el sistema.	X		La carencia en cuanto a la evaluación de los parámetros de seguridad influye de forma negativa al Monitor de Sitios Web cubano ya que causa un nivel de insatisfacción en los clientes, trayendo consigo perdidas al proyecto.
5	Se hace necesario la implementación de nuevas variables de seguridad para la evaluación de los sitios.	X		Como solución al problema de evaluación de parámetros de seguridad se hace necesaria la implementación de nuevas variables de seguridad para el Evaluador del Monitor de Sitios Web cubano.

Historias de Usuario.

Tabla 10: HU_3.

Historia de Usuario	
Número: HU_3	**Usuario:** Administrador
Nombre de la historia: Identificar la cabecera HSTS	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 1**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si un sitio implementa el encabezado HSTS, si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador	
Observaciones: En caso de que no se identifique el uso de la cabecera HSTS, no se evalúa el otro indicador relacionado con esta variable.	

Tabla 11: HU_4.

Historia de Usuario	
Número: HU_5	**Usuario:** Administrador
Nombre de la historia: Validar la cabecera HSTS	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 1**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si en el encabezado HSTS se encuentra presente el parámetro "*preload*", si lo utiliza se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 12: HU_5.

Historia de Usuario	
Número: HU_5	**Usuario:** Administrador
Nombre de la historia: Identificar la seguridad en las cookies	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 2**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si un sitio utiliza cookies, si no utiliza se le otorga una evaluación de 2, en caso de que si las utilice se evalúan los parámetros correspondientes a HttpOnly y Secure de ese indicador.	
Observaciones: En caso de que no se identifique el uso de cookies, no se evalúan el resto de los indicadores relacionados con esta variable.	

Tabla 13: HU_6.

Historia de Usuario	
Número: HU_6	**Usuario:** Administrador
Nombre de la historia: Validar el parámetro HttpOnly en las cookies	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 2**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite verificar si en el encabezado cookies se encuentra presente el parámetro “HttpOnly”, si se encuentra se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 14: HU_7.

Historia de Usuario	
Número: HU_7	**Usuario:** Administrador
Nombre de la historia: Validar el parámetro Secure en las cookies	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 2**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite verificar si en el encabezado cookies se encuentra presente el parámetro “Secure”, si se encuentra se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 15: HU_8.

Historia de Usuario	
Número: HU_8	**Usuario:** Administrador
Nombre de la historia: Identificar la cabecera Referrer Policy	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 3**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si un sitio implementa el encabezado Referrer Policy, si lo utiliza se le otorga una evaluación de 1 y se validan los indicadores relacionados con el mismo, sino se evalúa de 0 a ese indicador	
Observaciones: En caso de que no se identifique el uso de la cabecera Referrer Policy, no se evalúa el otro indicador relacionado con esta variable.	

Tabla 16: HU_9.

Historia de Usuario	
Número: HU_9	**Usuario:** Administrador
Nombre de la historia: Validar en la cabecera Referrer Policy los parámetros no-referrer, same-origin, strict-origin, strict-origin-when-cross-origin y no-referrer-when-downgrade	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 3**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite saber si en el encabezado Referrer Policy tiene presente alguno de los parámetros no-referrer, same-origin, strict-origin o strict-origin-when-cross-origin y no se encuentra el parámetro no-referrer-when-downgrade, en caso de cumplirse dicha condición se le otorga una evaluación de 1, en caso contrario se evalúa de 0 a ese indicador.	

Tabla 17: HU_10.

Historia de Usuario	
Número: HU_10	**Usuario:** Administrador
Nombre de la historia: Identificar procedencia de los recursos cargados	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 3**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana

Descripción: Permite verificar si todos los recursos que se cargan dentro del sitio son cargados por medio del protocolo seguro HTTPS, en caso de encontrarse todos los recursos cargados por esta vía se evalúa de 1, en caso contrario se evalúa de 0.

Tabla 18: HU_11.

Historia de Usuario	
Número: HU_11	**Usuario:** Administrador
Nombre de la historia: Validar Subresource Integrity	
Programador: José Luis Domínguez Echevarría	**Iteración Asignada: 3**
Prioridad en negocio: Alta	**Tiempo estimado:** 1 semana
Descripción: Permite verificar si cada uno de los recursos JavaScript externos al sitio son íntegros, es decir, no fueron modificados con respecto a su estado original, en caso de encontrarse de esta manera todos los recursos cargados se evalúa de 1, en caso contrario se evalúa de 0.	

Tarjetas CRS.

Tabla 19: Tarjeta CRC “csp”.

Tarjeta CRC	
Clase: csp	
Responsabilidades	**Colaboraciones**
Identificar la cabecera CSP	HttpURLConnection.

Validar la cabecera CSP	

Tabla 20: Tarjeta CRC “resource_load”.

Tarjeta CRC	
Clase: resource_load	
Responsabilidades	**Colaboraciones**
Identificar el protocolo utilizado para cargar recursos externos	Document.

Tabla 21: Tarjeta CRC “subresource_integrity”.

Tarjeta CRC	
Clase: subresource_integrity.	
Responsabilidades	**Colaboraciones**
Verificar si cada uno de los recursos JavaScript externos al sitio son íntegros	Document.

Tareas de Ingeniería.

Tabla 22: Tarea de Ingeniería Implementar variable que permite identificar la seguridad en las cookies.

Tarea de Ingeniería	
Número de la tarea: 8	**Nombre de la HU:** Identificar la seguridad en las cookies.
Nombre de la tarea: Implementar variable que permite identificar la seguridad en las cookies.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 7 días
Fecha de Inicio: 1 de Abril 2023	**Fecha fin:** 8 de Abril 2023
Programador responsable: José Luis Domínguez Echevarría.	
Descripción: Implementar la variable que permite saber si un sitio utiliza cookies, si no utiliza se le otorga una evaluación de 2, en caso de que si las utilice se evalúan los parámetros correspondientes a HttpOnly y Secure de ese indicador.	

Tabla 23: Tarea de Ingeniería Implementar variable que permite validar el parámetro HttpOnly en las cookies.

Tarea de Ingeniería	
Número de la tarea: 9	**Nombre de la HU:** Validar el parámetro HttpOnly en las cookies.
Nombre de la tarea: Implementar variable que permite validar el parámetro HttpOnly en las cookies.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 7 días
Fecha de Inicio: 8 de Abril 2023	**Fecha fin:** 15 de Abril 2023
Programador responsable: José Luis Domínguez Echevarría.	
Descripción: Implementar la variable que permite verificar si en el encabezado cookies se encuentra presente el parámetro “HttpOnly”, si se encuentra se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 24: Tarea de Ingeniería Implementar variable que permite Validar el parámetro Secure en las cookies.

Tarea de Ingeniería	
Número de la tarea: 10	**Nombre de la HU:** Validar el parámetro Secure en las cookies.
Nombre de la tarea: Implementar variable que permite validar el parámetro Secure en las cookies.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 7 días
Fecha de Inicio: 15 de Abril 2023	**Fecha fin:** 22 de Abril 2023
Programador responsable: José Luis Domínguez Echevarría.	
Descripción: Implementar la variable que permite verificar si en el encabezado cookies se encuentra presente el parámetro “Secure”, si se encuentra se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 25: Tarea de Ingeniería Implementar la funcionalidad que permita Identificar la cabecera Referrer Policy.

Tarea de Ingeniería	
Número de la tarea: 5	**Nombre de la HU:** Identificar la cabecera Referrer Policy
Nombre de la tarea: Implementar la funcionalidad que permita identificar la cabecera Referrer Policy	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 22 de Abril 2023	**Fecha fin:** 29 de Abril 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar la funcionalidad de la clase referrer_policy que permite conocer si un sitio implementa el encabezado Referrer Policy, si lo utiliza se le	

otorga una evaluación de 1 y se validan los indicadores relacionados con el mismo, sino se evalúa de 0 a ese indicador

Tabla 26: Tarea de Ingeniería Implementar la funcionalidad que permita Validar en la cabecera Referrer Policy los parámetros no-referrer, same-origin, strict-origin, strict-origin-when-cross-origin y no-referrer-when-downgrade.

Tarea de Ingeniería	
Número de la tarea: 7	**Nombre de la HU:** Validar en la cabecera Referrer Policy los parámetros no-referrer, same-origin, strict-origin, strict-origin-when-cross-origin y no-referrer-when-downgrade
Nombre de la tarea: Implementar la funcionalidad que permita validar en la cabecera Referrer Policy los parámetros no-referrer, same-origin, strict-origin, strict-origin-when-cross-origin y no-referrer-when-downgrade	
Tipo de tarea: Desarrollo	**Puntos estimados:** 7 días
Fecha de inicio: 29 de Abril 2023	**Fecha fin:** 6 de Mayo 2023
Programador responsable: José Luis Domínguez Echevarría	
Descripción: Implementar la funcionalidad de la clase referrer_policy que permite conocer si en el encabezado Referrer Policy tiene presente alguno de los parámetros no-referrer, same-origin, strict-origin o strict-origin-when-cross-origin y la no existencia del parámetro no-referrer-when-downgrade, en caso de cumplir con dicha condición se le otorga una evaluación de 1, sino se evalúa de 0 a ese indicador.	

Tabla 27: Tarea de Ingeniería Implementar variable que permite identificar procedencia de los recursos cargados.

Tarea de Ingeniería	
Número de la tarea: 11	**Nombre de la HU:** Identificar procedencia de los recursos cargados.
Nombre de la tarea: Implementar variable que permite identificar procedencia de los recursos cargados.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 7 días
Fecha de Inicio: 6 de Mayo 2023	**Fecha final:** 13 de Mayo 2023
Programador responsable: José Luis Domínguez Echevarría.	
Descripción: Implementar la variable que permite verificar si todos los recursos que se cargan dentro del sitio son cargados por medio del protocolo seguro HTTPS, en caso de encontrarse todos los recursos cargados por esta vía se evalúa de 1, en caso contrario se evalúa de 0.	

Tabla 28: Tarea de Ingeniería Implementar variable que permite validar Subresource Integrity.

Tarea de Ingeniería	
Número de la tarea: 12	**Nombre de la HU:** Validar Subresource Integrity.
Nombre de la tarea: Implementar variable que permite validar Subresource Integrity.	
Tipo de tarea: Desarrollo.	**Puntos estimados:** 7 días
Fecha de Inicio: 13 de Mayo 2023	**Fecha final:** 20 de Mayo 2023
Programador responsable: José Luis Domínguez Echevarría.	
Descripción: Implementar la variable que permite verificar si cada uno de los recursos JavaScript externos al sitio son íntegros, es decir, no fueron modificados con respecto a su estado original, en caso de encontrarse de esta manera se evalúa de 1, en caso contrario se evalúa de 0.	

Pruebas Funcionales

Caso de Prueba 3: Proceso de evaluación de la variable Referrer Policy.

Descripción general: Prueba a la funcionalidad de evaluación de la variable Referrer Policy.

Tabla 29: Caso de prueba proceso de evaluación de la variable Referrer Policy

Escenario	**Descripción**	**Evaluación**	**Respuesta del sistema**	**Flujo central**
EC 3.1 Realizar la evaluación al sitio web	El sistema realiza la evaluación de la variable Referrer Policy al sitio web	V	El sistema realiza	El sistema manda a evaluar el sitio web

https://www.facebook.com/	**https://www.facebook.com/** de manera correcta, indicando si se encuentra presente la cabecera y es válida.	Evaluación correcta.	la evaluación de la variable Referrer Policy de forma correcta, indicando que no se encuentra presente la cabecera y arroja como resultado una evaluación	**https://www.facebook.com/**

			de mal.	
EC 3.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	El sistema realiza la evaluación de la variable Referrer Policy al sitio web **https://observatory.mozilla.org/** de manera correcta, indicando si se encuentra presente la cabecera y es válida.	V Evaluación correcta.	El sistema realiza la evaluación de la variable Referrer Policy de forma correcta, indicando que se encuentra presente la cabec	**EC 1.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/**

			era y arroja como resultado una evaluación correcta al ser válida la misma.	
EC 3.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	El sistema realiza la evaluación de la variable Referrer Policy al sitio web **https://visuales.uclv.cu/** de manera correcta, indicando si se encuentra presente la cabecera y es válida.	V Evaluación correcta.	El sistema realiza la evaluación de la variable Referrer Policy de forma	El sistema manda a evaluar el sitio web **https://visuales.uclv.cu/**

			correcta, indicando que no se encuentra presente la cabecera y arroja como resultado una evaluación de mal.	

Descripción de las variables

Tabla 30: Descripción de las variables del caso de prueba proceso de evaluación de la variable Referrer Policy.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción

1	Tiene en la cabecera Referrer Policy.	Campo de texto.	No	Permite conocer si el sitio web tiene en la cabecera Referrer Policy.
2	Tiene válida la cabecera Referrer Policy.	Campo de texto.	No	Permite conocer si el sitio web tiene válida la cabecera Referrer Policy.

Caso de Prueba 4: Proceso de evaluación de la variable procedencia de los recursos cargados.

Descripción general: Prueba a la funcionalidad de evaluación de la variable procedencia de los recursos cargados.

Tabla 31: Caso de prueba proceso de evaluación de la variable procedencia de los recursos cargados.

Escenario	**Descripción**	**Evaluación**	**Respuesta del sistema**	**Flujo central**
EC 4.1 Realizar la evaluación al sitio web https://www.facebook.com	El sistema realiza la evaluación de la variable procedencia de los recursos cargados al sitio web **https://www.facebook.com/** de manera correcta, indicando que	V	El sistema realiza la evaluación de la variable	El sistema manda a evaluar el sitio web **https://www.facebook.com/**
		Evaluación correcta.		

	realiza la carga de cada recurso de forma segura.		procedencia de los recursos cargados de forma correcta, indicando que realiza la carga de cada recurso de forma segura.	
EC 4.2 Realizar la evaluación al sitio web	El sistema realiza la evaluación de la variable procedencia de los	V	El sistema realiza	El sistema manda a evaluar el sitio web **https://observatory.mozzilla.org/**

https://observatory.mozzilla.org	recursos cargados al sitio web **https://observatory.mozzilla.org/** de manera correcta, indicando que realiza la carga de cada recurso de forma segura.	Evaluación correcta.	la evaluación de la variable procedencia de los recursos cargados de forma correcta, indicando que realiza la carga de cada recurso de forma segura.	

EC 4.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	El sistema realiza la evaluación de la variable procedencia de los recursos cargados al sitio web **https://visuales.uclv.cu/** de manera correcta, indicando que realiza la carga de cada recurso de forma segura.	V Evaluación correcta.	El sistema realiza la evaluación de la variable procedencia de los recursos cargados de forma correcta, indicando que realiza la carga de cada recurso de forma	El sistema manda a evaluar el sitio web **https://visuales.uclv.cu/**

			segura .	

Descripción de las variables

Tabla 32: Descripción de las variables del caso de prueba proceso de evaluación de la variable procedencia de los recursos cargados.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Realiza carga de recursos externos.	Campo de texto.	No	Permite conocer si el sitio web realiza carga de recursos externos.
2	Realiza la carga de cada recurso externos de forma segura.	Campo de texto.	No	Permite conocer si el sitio web realiza la carga de los recursos externos mediante el protocolo HTTP.

Caso de Prueba 5: Proceso de evaluación de la variable Subresource-Integrity.

Descripción general: Prueba a la funcionalidad de evaluación de la variable Subresource-Integrity.

Tabla 33: Caso de prueba proceso de evaluación de la variable Subresource-Integrity.

Escenario	Descripción	Evaluación	Respuesta del	Flujo central

			sistema	
EC 5.1 Realizar la evaluación al sitio web https://www.facebook.com	El sistema realiza la evaluación de la variable Subresource-Integrity al sitio web **https://www.facebook.com/** de manera correcta, indicando que la procedencia de cara recurso cargado no se encuentra de forma íntegra.	V Evaluación correcta.	El sistema realiza la evaluación de la variable Subresource-Integrity de forma correcta, indicando que la procedencia de cara recurso cargado no se encuentra de	El sistema manda a evaluar el sitio web **https://www.facebook.com/**

			forma íntegra.	
EC 5.2 Realizar la evaluación al sitio web https://observatory.mozzilla.org	El sistema realiza la evaluación de la variable Subresource-Integrity al sitio web **https://observatory.mozzilla.org/** de manera correcta, indicando que la procedencia de cara recurso cargado se encuentra de forma íntegra.	V Evaluación correcta.	El sistema realiza la evaluación de la variable Subresource-Integrity de forma correcta, indicando que la procedencia de cara recurso cargad	El sistema manda a evaluar el sitio web **https://observatory.mozzilla.org/**

			o se encuentra de forma íntegra.	
EC 5.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	El sistema realiza la evaluación de la variable Subresource-Integrity al sitio web **https://www.facebook.com/** de manera correcta, indicando que la procedencia de cara recurso cargado se encuentra de forma íntegra.	V Evaluación correcta.	El sistema realiza la evaluación de la variable Subresource-Integrity de forma correcta, indicando que la procedencia de cara	El sistema manda a evaluar el sitio web **https:/visuales.uclv.cu/**

			recurso cargado se encuentra de forma íntegra.	

Descripción de las variables.

Tabla 34: Descripción de las variables del caso de prueba proceso de evaluación de la variable Subresource-Integrity.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Realiza carga de recursos externos.	Campo de texto.	No	Permite conocer si el sitio web realiza carga de recursos externos.
2	Realiza carga de recursos externos de forma íntegra.	Campo de texto.	No	Permite conocer si el sitio web realiza carga de recursos externos de forma íntegra.

Caso de Prueba 6: Proceso de evaluación de la variable Seguridad en las Cookies.

Descripción general: Prueba a la funcionalidad de evaluación de la variable Seguridad en las Cookies.

Tabla 35: Caso de prueba proceso de evaluación de la variable Seguridad en las Cookies.

Escenario	Descripción	Evaluación	Respuesta del sistema	Flujo central
EC 6.1 Realizar la evaluación al sitio web https://www.facebook.com	El sistema realiza la evaluación de la variable Seguridad en las Cookies al sitio web **https://www.facebook.com/** de manera correcta, indicando que no se aseguran las cookies.	I Evaluación Invalida.	El sistema realiza la evaluación de forma incorrecta de la variable Seguridad en las Cookies, indicando que no se	El sistema manda a evaluar el sitio web **https://www.facebook.com/**

			gestionan las cookies de manera correcta.	
EC 6.2 Realizar la evaluación al sitio web https://observatory.mozzilla.org	El sistema realiza la evaluación de la variable Seguridad en las Cookies al sitio web **https:/observatory.mozzilla.org/** de manera correcta, indicando que si se gestionan las cookies y que los parámetros HttpOnly y Secure estén activos.	V Evaluación correcta.	El sistema realiza la evaluación de forma correcta de la variable Seguridad en las Cookies, indicando	El sistema manda a evaluar el sitio web **https:/observatory.mozzilla.org/**

			que si se gestionan las cookies y que los parámetros HttpOnly y Secure estén activos.	
EC 6.3 Realizar la evaluación al sitio web https://visuales.uclv.cu	El sistema realiza la evaluación de la variable Seguridad en las Cookies al sitio web **https:/visuales.uclv.cu/** de manera correcta, indicando que no se gestionan las cookies, por tanto no pueden existir	V Evaluación correcta.	El sistema realiza la evaluación de forma correcta de la	El sistema manda a evaluar el sitio web **https:/visuales.uclv.cu/**

	riesgos en cuanto a este parámetro.		variable Seguridad en las Cookies, indicando que no se gestionan las cookies, por tanto no pueden existir riesgos en cuanto a este parámetro.	

Descripción de las variables

Tabla 36: Descripción de las variables del caso de prueba proceso de evaluación de la variable Seguridad en las Cookies.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Realiza la gestión de cookies.	Campo de texto.	No	Permite conocer si el sitio web realiza la gestión de cookies.
2	Se encuentre activo el parámetro HttpOnly dentro de la gestión de las cookies.	Campo de texto.	No	Permite conocer si en el sitio web se encuentre activo el parámetro HttpOnly dentro de la gestión de las cookies.
3	Se encuentre activo el parámetro Secure dentro de la gestión de las cookies.	Campo de texto.	No	Permite conocer si en el sitio web se encuentre activo el parámetro Secure dentro de la gestión de las cookies.

Pruebas unitarias.

Tabla 37: Evaluación del método ReferrerPolicy de la clase referrer_policy.

Escenario	Respuesta del sistema	Respuesta esperada
EC 3.1 Realizar la evaluación al sitio web https://www.facebook.com/	0	0

EC 3.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	2	2
EC 3.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	0	0

Tabla 38: Evaluación del método ResourceLoad de la clase resource_load.

Escenario	Respuesta del sistema	Respuesta esperada
EC 3.1 Realizar la evaluación al sitio web https://www.facebook.com/	1	1
EC 3.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	1	1

EC 3.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	1	1

Tabla 39: Evaluación del método SoubresourceIntegrity de la clase soubresource_integrity.

Escenario	Respuesta del sistema	Respuesta esperada
EC 3.1 Realizar la evaluación al sitio web https://www.facebook.com/	0	0
EC 3.2 Realizar la evaluación al sitio web https://observatory.mozilla.org/	1	1
EC 3.3 Realizar la evaluación al sitio web https://visuales.uclv.cu/	1	1

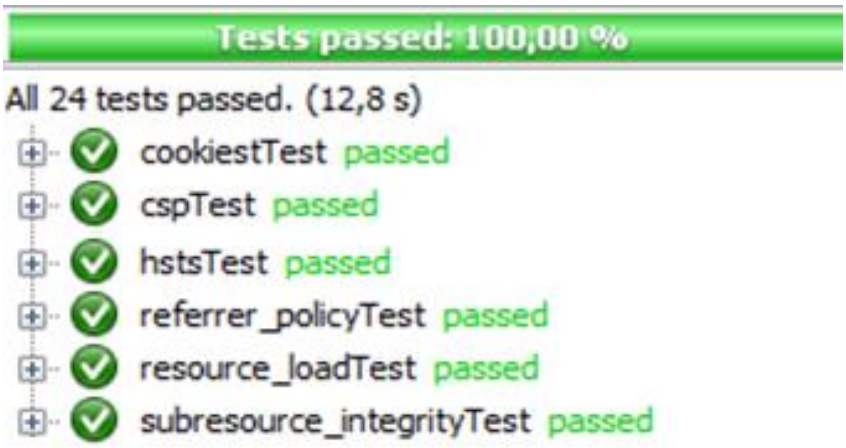


Figura 4: Resultados de las pruebas unitarias. Fuente: Elaboración propia.