**OFICINA DE POSTGRADOS**

**Tema:**

# IMPLEMENTACIÓN DE UNA SOLUCIÓN WEB ISOLATION PARA LA DISMINUCIÓN DE AMENAZAS EN EQUIPOS CLIENTES

**Proyecto de investigación previo a la obtención del título de Magister en Ciberseguridad**

**Línea de Investigación:**

Protección de datos y comunicaciones

**Autora:**

Ericka Liseth Guanoluisa Paredes

**Director:**

Mg. Paul Fernando Bernal Barzallo

**Ambato – Ecuador**

**JULIO 2021**

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**SEDE AMBATO**

**HOJA DE APROBACIÓN**

**Tema:**

**IMPLEMENTACIÓN DE UNA SOLUCIÓN WEB ISOLATION PARA LA DISMINUCIÓN DE AMENAZAS EN EQUIPOS CLIENTES**

**Línea de Investigación:**

Protección de datos y comunicaciones

**Autora:**

Ericka Liseth Guanoluisa Paredes

Firmado electrónicamente por:
PAUL FERNANDO BERNAL BARZALLO

Paul Fernando Bernal Barzallo**,** Mg. f._____

**CALIFICADOR**

Darío Javier Robayo Jacome, MSc. f._____

**CALIFICADOR**

Galo Mauricio López Sevilla, MSc. f._____

**CALIFICADOR**

Juan Carlos Acosta Teneda. Ing. Mg. f._____

**COORDINADOR DE LA OFICINA DE POSGRADOS**

Hugo Rogelio Altamirano Villaroel, Dr. f._____

**SECRETARIO GENERAL PUCESA**

Ambato – Ecuador

JULIO 2021

**DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD**

Yo: **ERICKA LISETH GUANOLUISA PAREDES**, con CC. **0503510414** autora del trabajo de graduación intitulado: **"IMPLEMENTACIÓN DE UNA SOLUCIÓN WEB ISOLATION PARA LA DISMINUCIÓN DE AMENAZAS EN EQUIPOS CLIENTES"**, previa a la obtención del título profesional de MAGISTER EN CIBERSEGURIDAD, en la OFICINA DE POSGRADOS.

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Ambato, julio 2021

Firmado electrónicamente por:
**ERICKA LISETH GUANOLUISA PAREDES**

ERICKA LISETH GUANOLUISA PAREDES

CC. 0503510414

**AGRADECIMIENTO**

A mis apreciados padres, por su amor y cariño incondicional.

A mi esposo por ser mi fuerza y apoyo, por motivarme a cumplir todos los objetivos y metas propuestas.

A mi tutor, el Mg. Paul Bernal por la experiencia, conocimiento y tiempo compartido en la realización del presente trabajo.

*Ericka*

## DEDICATORIA

Como muestra de profundo amor, dedico el presente trabajo a mi esposo Miguel, por ser quién día a día me motiva y apoya para alcanzar los objetivos de mi vida profesional y familiar.

*Ericka*

**RESUMEN**

Dado que, en el Ecuador el 95% de las unidades productivas son pequeñas y medianas empresas, que no poseen el presupuesto necesario para implementar una herramienta de aislamiento web por su elevado costo, el presente trabajo de investigación tiene como objetivo implementar un sistema de Web *Isolation* económico que permita a las empresas mencionadas el acceso a esta herramienta como mecanismo de seguridad frente a las amenazas web en los equipos clientes. Como metodología de desarrollo se siguió Scrum y al ser un proyecto con un enfoque experimental, dado que, se requiere establecer comparaciones del antes y después de la implementación de la solución planteada, se considera como metodología de investigación a la experimental. Para el desarrollo de la solución, se utilizaron tecnologías como *Docker, Docker Compose, Voodoo, Zombie Lord, Dart, Flutter, Java, Spring Boot, Firebase, DNSMasq* entre otras. Con la aceptación de un error igual a 0.05 equivalente al 95% de acierto, con un p valor igual a 0.03125, con la solución propuesta, se obtiene una herramienta *open source*, sin costo de instalación ni mantenimiento, capaz de mitigar al 100% las amenazas experimentadas, sin perjudicar la velocidad de navegación de los equipos clientes.

**PALABRAS CLAVES:** Aislamiento Web, Mitigación Vulnerabilidades, Aislamiento de Navegador.

**ABSTRACT**

Since, at that in Ecuador 95% of the production units are small and medium-sized companies, which don't have the necessary budget to implement a web isolation tool due to its expensive cost, the present research work has as general objective, the implementation of an Economic Web Isolation system that allows this companies companies access to this kind of tools as a security mechanism against web threats on client computers. Scrum was followed as a development methodology and being a project with an experimental approach, since it is required to establish comparisons before and after the implementation of the proposed solution, it is considered an experimental research methodology. For the development of the solution, technologies such as Docker, Docker Compose, Voodoo, Zombie Lord, Dart, Flutter, Java, Spring Boot, Firebase, DNSMasq among others were used. With the acceptance of an error equal to 0.05 equivalent to 95% of success, with a p value equal to 0.03125, with the proposed solution, an open source tool is obtained, with no installation or maintenance costs, capable of mitigating 100% of the threats experimented, without affecting the browsing speed of client computers.

**KEYWORDS:** Web Isolation, Vulnerability mitigation, Browser Isolation.

**ÍNDICE GENERAL**

**PRELIMINARES**

## ÍNDICE DE TABLAS

## ÍNDICE DE ILUSTRACIONES

## INTRODUCCIÓN

Los ataques basados en el navegador, son la principal fuente de amenazas a los usuarios, es así, la navegación web en un entorno empresarial e institucional una tarea con muchos riesgos, dado que, los atacantes utilizan los sitios web para la propagación de amenazas como*: spam, spyware, adware,* ataques de *phishing*, distribución de códigos maliciosos, troyanos, scripts de archivos portables ejecutables, robar información confidencial de los equipos clientes y amenazas web dirigidas deliberadamente a los sistemas operativos más comunes como: android y Windows (Kaspersky, 2019), además, a aplicaciones como: java, adobe reader o internet explorer entre otras, lo cual conducen a graves consecuencias tanto financieras como jurídicas.

Si bien, el uso de antivirus, firewalls o actualizaciones de seguridad de los navegadores, se ha podido hacer frente a infecciones de software malicioso instalable, las mismas que son distribuidas a través de enlaces engañosos como mensajes de correo electrónico, mensajes instantáneos o sitios web que infectan los equipos clientes, para los expertos en seguridad de McAFee (2020) aún no se garantiza la totalidad de la seguridad, dado que, es imprescindible acceder a sitios web.

En relación a esta necesidad MacDonald (2016) del grupo Gartner en su reporte *It's Time to Isolate Your Users from the Internet Cesspool with Remote Browsing* predijo que para el año 2018 la mitad de las empresas comenzarán a aislar activamente la navegación web de sus equipos, tecnología denominada *Web Isolation*.

Actualmente, existen diversas soluciones de *Web Isolation*, como: ViewFinder cuyo costo de aislamiento por sitio es de $100 dólares mensuales (dosyago, 2020), Cigloo ofrece esta solución desde $20 mensual por usuario (Sourceforge, 2019), así también, Fortinet FortiIsolator presenta una solución basada en *hardware* por un costo de $11,792.86 (Avfirewalls, 2020).

Del párrafo anterior se evidencia que existen problemas asociados en la tecnología de *Web Isolation* como: el costo, una solución implica un gasto mensual de entre 15 y 50

dólares por equipo cliente situación que limita a las empresas sin un fuerte fondo para ciberseguridad; y la escalabilidad, dado que, un crecimiento en las necesidades implica un crecimiento exponencial en los gastos de operación por el elevado costo individual.

En Ecuador, según Ron & Sacoto (2017) el 95% de las unidades productivas son generadas por las pequeñas y medianas empresas, mismas que por su tamaño generalmente no poseen un fondo para ciberseguridad, lo que dificulta el acceso a las soluciones de *Web Isolation* privativas existentes.

En este sentido, el problema científico de la investigación se define: ¿Cómo disminuir el costo de la implementación de una herramienta de *Web Isolation* para el aseguramiento de la navegación de los equipos clientes?, y la hipótesis de trabajo se establece como: La implementación de una herramienta accesible de Web *Isolatio*n, permitirá disminuir los costos del aseguramiento de los equipos clientes.

Es así que el presente trabajo de investigación tiene como objetivo general implementar un sistema de *Web Isolation* económico que permita a las pequeñas y medianas empresas el acceso a esta herramienta como mecanismo de seguridad frente a las amenazas web en los equipos clientes, para el cumplimiento se plantea los siguientes objetivos específicos:

- Fundamentar teóricamente el uso de las herramientas de Web *Isolation* existentes, para el desarrollo de la investigación.
- Definir los requerimientos mínimos para la herramienta de Web *Isolation* en la disminución de los costos del aseguramiento de la navegación de los equipos clientes.
- Diseñar la solución Web *Isolation* como mecanismo de seguridad frente a las amenazas web en los equipos clientes.

Una vez definidos los elementos de la solución Web *Isolation*, para el desarrollo se analiza la metodología que se adapte mejor a la propuesta, para lo cual previo análisis se determina que el proyecto es ágil y por la naturaleza de la propuesta podría llevar consigo cambios repentinos o aparecer nuevas necesidades a cubrir a corto plazo, por

lo que establece que la metodología Scrum cumple con lo que se requiere en la implementación de la propuesta, por otra parte se al ser un proyecto con un enfoque experimental y se requiere establecer comparaciones del antes y después de la implementación de la solución de *Web Isolation*, se considera como metodología de investigación a la experimental.

Se considera todo lo mencionado anteriormente, además, de la situación social de los países en vías de desarrollo, quienes prefieren invertir en la parte operativa más que en la seguridad de la información, se hace visible la necesidad que justifica la presente investigación, pues, al entregar a las Micro Pequeñas y Medianas Empresas (PYME) una herramienta de *software* libre gratuita, su uso podrá masificarse y mejorar así la seguridad sin una fuerte inversión económica, es importante, también, aclarar que es un sistema de prevención de ataques, no orientado a la corrección de incidentes de seguridad, por cuanto usarían en ambientes previamente asegurados para poder conservarlos en ese estado.

Para mantener la privacidad de la información, que es un tema muy demandado actualmente, se recomienda que el servidor de Web Isolation se encuentre desplegado de manera local en la infraestructura informática a proteger, pero, con la filosofía del software libre, se espera que alguna institución se convierta en hospedera oficial de la solución y la comparta como servicio a las pyme interesadas.

# CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA

## 1.1. Peligros en la navegación web

Los navegadores web al ser una de las aplicaciones más utilizadas, son las más difíciles de proteger debido a su complejidad, dado que, llevan a cabo la acción más peligrosa que realiza un programa informático, que es descargar código que no es de confianza y ejecutarlo directamente en el equipo de un usuario por lo que los ataques basados en navegador sean el principal vector de amenaza para que los atacantes se dirijan a los usuarios. Según Osterman (2018) en su informe *Why You Should Seriously Consider Web Isolation Technology*, el 60% de las organizaciones se han infectado con malware como resultado de la navegación web.

En concordancia con lo mencionado Zscaler (2020) menciona:

> "La necesidad de una conexión persistente a Internet está exponiendo a las organizaciones a un mayor riesgo, la mayoría de los ciberataques se dirigen a los usuarios a través de sus navegadores, publicando publicidad maliciosa (anuncios maliciosos), cebo de clics que puede conducir a contenido malicioso, troyanos basados en navegador y más. Una vez que un navegador está conectado a un sitio, les da a los ciberdelincuentes una puerta abierta a la máquina del usuario y, muy probablemente, a su red."

Esta es una realidad latente, a pesar de todos los esfuerzos que constantemente realizan los fabricantes y desarrolladores de los principales navegadores como Chrome y Mozilla Firefox.

Bajo este contexto, también, se considera que en una organización los colaboradores dedican una cantidad significativa de tiempo a actividades en la web no relacionadas al trabajo, concuerda con esto Osterman (2018), al estimar que los usuarios dedican al menos tres horas al día en la navegación web no laboral, es así que cada vez que en un equipo cliente se navega en la web sin el uso de restricciones, se expone potencialmente a amenazas incalculables para la organización como se evidencia en la tabla 1:

*Tabla 1: Amenazas Web*

| Amenaza Web | Descripción | Casos Reales |
|---|---|---|
| Filtración de Información | Mediante el uso de *phishing* o malware distribuido por medio de correo electrónico o cualquier sistema web el atacante logra convencer a la víctima para que interactúe con un determinado enlace que podría permitirle al atacante, entre otras cosas: el robo de identidad, datos, etc. | **Filtración de datos de Target Corp en 2013:** Dos días antes del Black Friday, se robaron 110 millones de registros de tarjetas de crédito y débito de los clientes, los detalles específicos del ataque no han sido publicados, pero se tiene comprobado que una campaña de *phishing* por correo electrónico a uno de sus proveedores fue el origen del ataque. La empresa informó que este evento le costó un total de $ 290 millones de dólares.(Paúl, 2016)<br><br>**Robo de datos a las cadenas de restaurantes Chipotle, Arby´s y Chili´s en 2017:** Las cadenas recibierón correo electrónico que incluían software malicioso adjunto, se instaló silenciosamente en los equipos lo que permitió con ello aceso a las redes internas de estas empresas y consiguieron robar más de 15 millones de registros de tarjetas de crédito pertenecientes a los clientes.(Valinsk, 2018) |
| Ataques de *ransomware* | Es un tipo de malware que tiene como objetivo el secuestro de datos, obliga a las personas u organizaciones el pago por el rescate de la información. Se complementan con campañas de *phishing*, a través de las cuales consiguen que su archivo malicioso llegue a un dispositivo final, además, estos archivos infectados, se distribuyen también por sitios web maliciosos con descargar automáticas. | **Bad Rabbit 2017:** Según Perekalin( 2017)**,** se trata de un ataque no deseado: las víctimas descargaron un instalador de Adobe Flash falso de los sitios web infectados y ejecutaron manualmente el archivo .exe. Entre las víctimas confirmadas del malware se menciona a los medios de comunicación rusos: agencia de noticias Interfax y Fontanka.ru. Los criminales detrás del ataque Bad Rabbit pedían 0.05 bitcoin como rescate (aproximadamente $ 280 dólares).<br><br>**Jigsaw 2016**: Fue diseñado para propagarse mediante archivos adjuntos maliciosos en correos electrónicos no deseados. Este malware cifra los archivos y exige un rescate para recuperarlos, comienza una cuenta regresiva. Si el rescate de $ 150 no se paga dentro de la primera hora, se elimina un archivo. A medida que pasa el tiempo, se elimina más de un archivo cada hora, ese número aumenta cada vez que se reinicia el temporizador de 60 minutos y cada vez que se reinicia el programa, se eliminan hasta 1000 archivos.(Norton, 2020) |
| Malware bancario | Es una amenaza que infecta un equipo con el fin de monitorear | **Dridex:** Es un troyano que va evolucionado, apareció en 2011, pero su mayor impacto |

| | | |
|---|---|---|
| | los inicios de sesión de cuentas bancarias para robar las credenciales de acceso mediante las técnicas: *keylogger* o *webinjects,* diseñadas específicamente para instituciones bancarias. | fue en el año 2015 en el cual causó daños estimados en más de $ 40 millones en entidades financieras. Mediante el uso de *webinjects* pudo robar dinero e infectar a dispositivos USB. Según el informe de Slepogin (2017), en la cuarta versión de Dridex a inicios del 2017, se detectó esta amenaza en varios países europeos, en donde el Reino Unido representó el 60% de las víctimas, seguido de Alemania y Francia. |

Fuente: elaboración propia

### 1.2. Web Isolation

Es un enfoque de la Ciberseguridad, que pretende separar la navegación realizada por el hardware de la información con la que interactúa el cliente final, por lo cual proporciona un nivel adicional protección al usuario, lo que reduce la superficie de ataque, en resumen, el usuario final, simplemente recibe una renderización del contenido, mientras que un navegador remoto interactúa con el origen de la información.

Esta tecnología funciona de la siguiente manera:

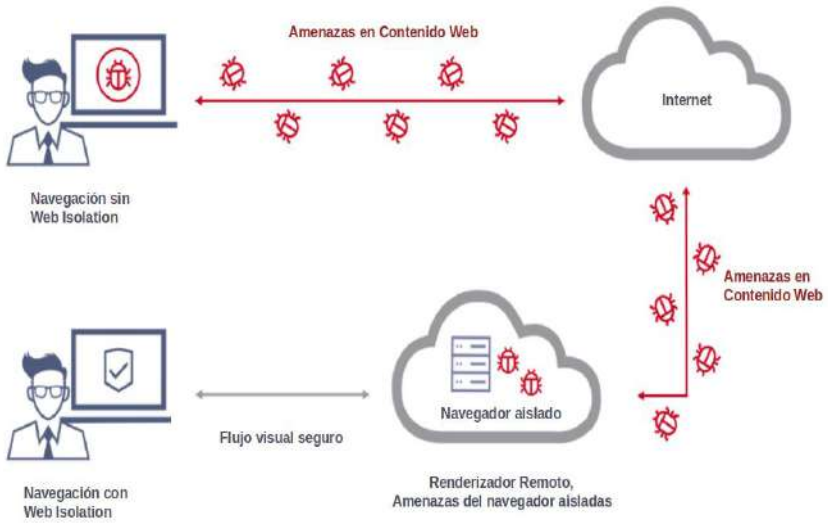
1.- El cliente inicia una petición web.

2.-De acuerdo a las políticas internas de la organización se determina el paso o no del contenido del sitio web solicitado y si este sería o no aislado.

3.- Si se determina que el sitio web será asilado, este es ejecutado dentro de un contenedor o servidor seguro que estaría en la organización o alojado en la nube, devuelve de esta manera información procesada de manera segura a los navegadores de los equipos clientes.

4.- En el equipo cliente, se muestra un flujo visual en tiempo real del sitio web que está es ejecutado en el servidor.

*Ilustración 1: Funcionalidad Web Isolation*




Fuente: adaptado a partir de McAFee (2020)

Entre los beneficios de la tecnología *Web Isolation*, se menciona que: al reducir el riesgo de una infección de malware basada en la web, reduce considerablemente el costo de reparaciones, así como el tiempo invertido en las mismas; ofrece protección al usuario de correos electrónicos, sitios web, enlaces, descargas y anuncios maliciosos.

### 1.3. Open Source

Con la finalidad de cumplir con el objetivo general del proyecto en el cual se plantea desarrollar una solución de *Web Isolation* que sea económicamente accesible para las pequeñas y medianas empresas, se hace uso de herramientas *open source* se justifica su uso por la principal característica de distribuir libremente el software mediante una licencia de código abierto que permite al usuario final analizarlo y modificarlo para adaptarlo a sus necesidades, además, de promover valores como la colaboración, el compañerismo, el acceso de todos a la tecnología y la igualdad de oportunidades, que son los pilares de la filosofía *open source,* así como lo menciona (Red Hat, 2021).

## 1.4. Metodologías ágiles

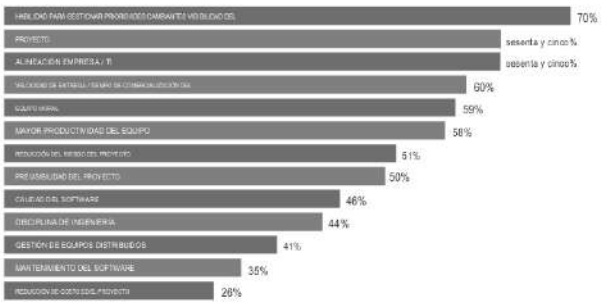
Los métodos de desarrollo ágil fueron formalizados en el Manifiesto para el Desarrollo de Software Ágil (Beck et al., 2001), conocidos en un inicio como métodos ligeros, por la contraposición a los métodos tradicionales, pesados.

Entre las diferencias que destacan en la comparación de métodos (ágiles) con los tradicionales (o pesados) se tiene que generan menos documentación. Los ágiles se enfocan principalmente en la legibilidad del código fuente, para que el mismo no requiera una documentación adicional que explique su funcionamiento, en base a lo mencionado se pone de manifiesto dos diferencias más significativas.

Primero, los métodos tradicionales, tienen una marcada tendencia a la excesiva planificación del flujo de desarrollo de software, lastimosamente, estas planificaciones no funcionan correctamente en caso de existir cambios, por otro lado, los métodos ágiles al ser más adaptables que predictivos, asumen los cambios como una acción impredecible e inevitable y se prepara para recibirlo, implanta de esta manera una filosofía de pronta adaptación a los cambios. Segundo, el agilismo prioriza a las personas por sobre el proceso, lo cual, motiva la naturaleza del comportamiento humano en lugar de tratar de controlarlo.

Según, Digital.ai (anteriormente CollabNetVersionOne)(2020) en su reporte *14th Annual State of Agile Report*, entre las principales razones para adoptar una metodología ágil se tiene la aceleración en la entrega de software con un 71% y la mejora en la capacidad de gestionar las prioridades cambiantes en un 63%.

*Ilustración 2: Razones para uso de metodologías ágiles*



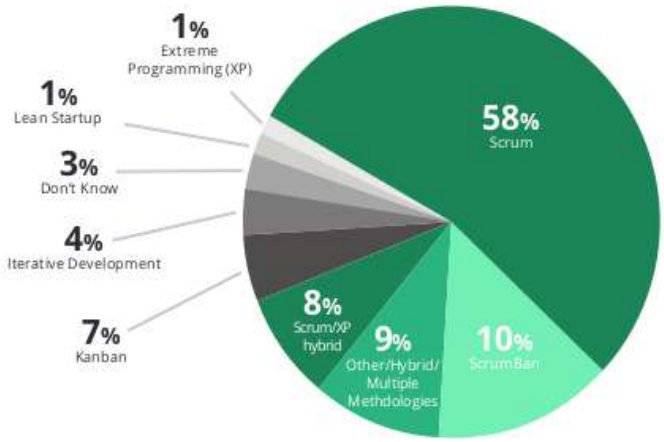

Fuente: adaptado a partir de Digital.ai ( 2020)

### 1.4.1. Scrum

Es un marco de trabajo que sigue los fundamentos establecidos en el manifiesto ágil, desarrollado a principios de la década de 1990. Se basa en el empirismo, dado que, pretende que el conocimiento sea causa de la experiencia, así como de la observación y toma de decisiones, además, incluye el pensamiento Lean, mismo que trata de optimizar los recursos lo que promueve el enfoque y la optimización del tiempo (Schwaber & Sutherland, 2020).

Con la finalidad de generar una apropiada solución de *Web Isolation* que se adapte a las necesidades del proyecto, se revisa las metodologías disponibles y se tiene que Scrum sugiere un proceso simple pero funcional para desarrollar el proyecto. Para afianzar la selección de este marco de trabajo, se menciona los resultados del informe de Digital.ai, en el que Scrum y sus variantes relacionadas siguen consideradas las metodologías ágiles más comúnmente utilizadas en las organizaciones, y tienen un alto margen de éxito en los proyectos que han seguido este principio.

*Ilustración 3: Metodologías ágiles más utilizadas*




Fuente: adaptado a partir de Digital.ai ( 2020)

La versión de Scrum a utilizarse en el desarrollo será la más actual, en este caso la publicación oficial de noviembre 2020, la guía sobre su aplicación se obtiene en el siguiente enlace: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf

# CAPÍTULO II. DISEÑO METODOLÓGICO

## 2.1. Metodología de desarrollo

Para el desarrollo de la solución propuesta, se establece como marco de trabajo a Scrum, a continuación, se describe detalladamente los procesos realizados en cada uno de los aspectos fundamentales propuestos:

### 2.1.1. Scrum Team

Se ha definido la participación de dos personas y tres responsabilidades para la implementación planteada:

*Tabla 2: Scrum Team*

| Responsabilidad | Persona | Descripción |
|---|---|---|
| *Product Owner* | Paul Bernal | Conoce y define las funcionalidades requeridas para la implementación planteada. |
| *Developers* | Ericka Guanoluisa | Encargada de la implementación planteada por el *Product Owner*. |
| Scrum *Master* | Ericka Guanoluisa | Asegura el seguimiento de la metodología al orientar a los miembros del equipo en ser autogestionados y multifuncionales. |

Fuente: elaboración propia

### 2.1.2. Artefactos y Eventos

#### 2.1.2.1. Product backlog

Como parte del proceso de desarrollo con Scrum se empezó con la lista de funcionalidades requeridas y priorizadas por el p*roduct owner*, responsabilidad que fue tomado por el director del presente trabajo de titulación quien tiene amplia experiencia en servicios informáticos avanzados.

*Tabla 3: Product backlog*

| N° | HISTORIA DE USUARIO | ESTADO | SPRINT |
|---|---|---|---|
| 1 | Definición de la arquitectura de la solución *web isolation*. | Terminado | 1 |
| 2 | Preparación del ambiente de desarrollo. | Terminado | 1 |
| 3 | Despliegue de la infraestructura para pruebas | Terminado | 2 |
| 4 | Implementación de servidor DnsMasq | Terminado | 2 |
| 5 | Implementación de un servidor de isolación | Terminado | 3 |
| 6 | Plataforma web para la gestión de sitios a aislar | Terminado | 4 |

| | | | |
|---|---|---|---|
| 7 | Implementación de Usuarios y Roles para el control de acceso a la web | Terminado | 4 |
| 8 | Sesiones independientes en el servidor de isolación | Terminado | 5 |
| 9 | Bloqueo de creación nuevas pestañas dentro del navegador aislado | Terminado | 5 |
| 10 | Gestión de contenido multimedia. | Terminado | 6 |
| 11 | Estadísticas de comportamiento de la navegación aislado. | Terminado | 6 |
| 12 | Implementación de Reverse Proxy para redirección del puerto 80 al del API de *Web Isolation* | Terminado | 7 |
| 13 | Reemplazar reverse proxy por un software propio encargado de la redirección | Terminado | 8 |
| 14 | Diseño de una solución para aislar sitios https | Terminado | 9 |

Fuente: elaboración propia

El *product backlog* se llenó a partir de las historias de usuario, en las que se coloca pequeñas descripciones de los requerimientos del cliente, al redactarlas se considera que el título sea claro, que la explicación sea suficientemente detallada, así como los criterios de aceptación sean puntuales y amplios. En la tabla 4 se presenta la historia de usuario realizada implementación de la solución *Web Isolation* para la disminución de amenazas en equipos clientes. Todas las historias de usuario generadas siguen el mismo formato y se las encuentran en el Anexo 1: Historias de Usuario del presente documento.

*Tabla 4: Historia de Usuario 7*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 7 | **Sprint:** 4 |
| **Nombre de la Historia:** Implementación de Usuarios y Roles para el control de acceso a la web | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 27/10/2020 | **Fecha Fin:** 03/11/2020 |
| **Descripción:**<br>**Contexto** Dado que la configuración de la herramienta es un proceso administrativo, se controla el acceso al mismo únicamente a ciertos usuarios, para lo cual se requiere la implementación de un proceso de autenticación y control de acceso a la web de administración<br>**Como Administrador de la herramienta** necesito controlar mi acceso a la web de administración **para** prevenir modificaciones no autorizadas.<br>**Criterios de aceptación**<br>1. Únicamente un usuario con inicio de sesión correcto accedería al listado de sitios<br>2. En caso de un usuario o contraseña incorrectos mostrar el mensaje “Credenciales Incorrectas”<br>3. En caso de intentar navegar por url a algún destino con evasión de la autenticación sería redireccionado a la página de inicio de sesión si no tiene una sesión válida iniciada | |

Fuente: elaboración propia

**Sprint**

Para el desarrollo de las historias del *backlog*, se dividirá el trabajo en sprints, scrum recomienda que su duración no supere los 30 días, es así que, para el desarrollo del proyecto, por cuestiones de tiempo y horario del equipo, se establecen Sprint de 15 días calendario; en total se desarrollaron 7 sprints con los rangos de fechas especificados en la Tabla 5.

#### 2.1.2.2. Sprint *planning*

Se realiza al inicio de cada sprint para determinar las funcionalidades a implementar en ese lapso de tiempo, es importante la definición de un Sprint *Goa*l o meta del sprint que será el punto de foco del equipo durante esta iteración, las historias se alinean a este objetivo y cualquier acuerdo o decisión estará a favor de lograr el mismo.

*Tabla 5: Sprint planning*

| Sprint | Historia de Usuario | Fecha de inicio | Fecha de Fin | Sprint *Goal* |
|---|---|---|---|---|
| 1 | 1,2 | 07/09/2020 | 16/09/2020 | Tener el entorno preparado para iniciar la Implementación. |
| 2 | 3,4 | 17/09/2020 | 02/10/2020 | Tener un servidor remoto de resolución DNS. |
| 3 | 5 | 03/10/2020 | 18/10/2020 | Poder visualizar un sitio web aislado. |
| 4 | 6,7 | 19/10/2020 | 03/11/2020 | Poder gestionar los sitios web aislados. |
| 5 | 8,9 | 04/11/2020 | 19/11/2020 | Mejorar la seguridad de la herramienta de isolación. |
| 6 | 10,11 | 20/11/2020 | 05/12/2020 | Mejorar las funcionalidades de la herramienta de isolación. |
| 7 | 12 | 06/12/2020 | 21/12/2020 | Producto Integrado listo para producción. |
| 8 | 13 | 01/03/2021 | 0503/20 | Agregar soporte https |
| 9 | 14 | 08/03/2021 | 12/03/2021 | Agregar soporte https |

Fuente: elaboración propia

#### 2.1.2.3. Sprint backlog

Durante el *planning,* se toman las tareas del *product backlog* y según las capacidades del Scrum *Team*, se arma un Sprint *Backlog*, cada tarea al ser tomada de aquí durante el Sprint en curso es desarrollada y genera Tareas Relacionadas, como ejemplo a continuación se listan las tareas de la historia de usuario 7:

1. Agregar dependencia spring *security* en el *back end*

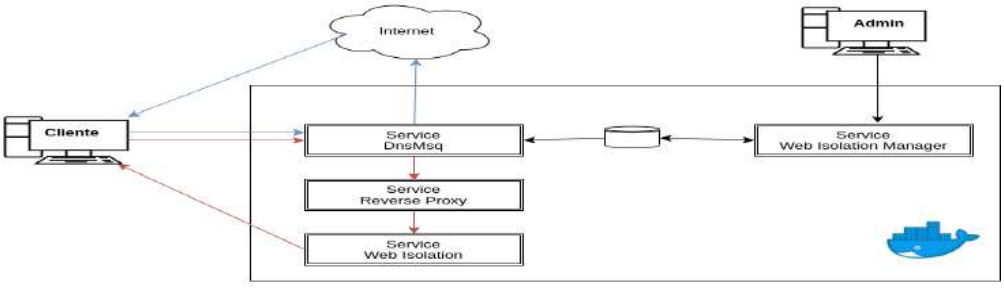
2. Agregar un *provider* de *tokens* jwt en el servidor *back end*
3. Crear pantalla de autenticación en el *front end*
4. Implementar función de autenticación contra el servidor *back end*
5. Establecer un tiempo de caducidad del *token*
6. Agregar un interceptor en el *back end* para verificar la validez del *token*, en caso de ser inválido retornará un error 403 *Unauthorized*
7. Implementar un local *storage* en el *front end* para almacenar el *token* jwt
8. Implementar un interceptor en el *front end* para adjuntar el *token* en cada petición al back (campo Authorization)
9. Implementar un interceptor en el front end para redireccionar a /login cualquier intento de petición que retorne un error 403 *Unauthorized* por parte del *back end*

#### 2.1.2.4. Increment

Como resultado de la historia desarrollada se obtiene una nueva funcionalidad para el producto potencialmente desplegable en producción, la suma de todas las funcionalidades desarrolladas en el Sprint se la conoce en Scrum como incremento. Se generó un total de 7 incrementos los mismos que se detallan a continuación.

*Incremento 1:* Con los criterios de aceptación de la historia de usuario 1 se utilizó para el desarrollo del diagrama de la arquitectura la herramienta en línea draw.io, se considera que la arquitectura estaría compuesta exclusivamente por herramientas de software libre que permitan la escalabilidad. Como se muestra en la ilustración 4.

*Ilustración 4: Arquitectura del proyecto*




Fuente: elaboración propia

*Incremento 2:* Se obtuvo el servidor con docker y docker compose sobre, los cuales, se ejecutarán las diferentes partes de la solución, de la misma manera sobre esta infraestructura se despliega el servicio dnsmasq que será el encargado de la resolución de dominios en la red y redirección al api de isolación.

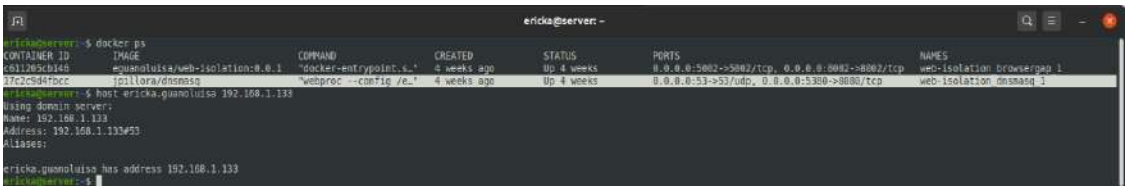
*Ilustración 5: Docker y docker-compose instalado*


Fuente: elaboración propia

*Ilustración 6: Archivo de instrucciones para docker-compose*


Fuente: elaboración propia

*Ilustración 7: Servicio dnsmasq*




Fuente: elaboración propia

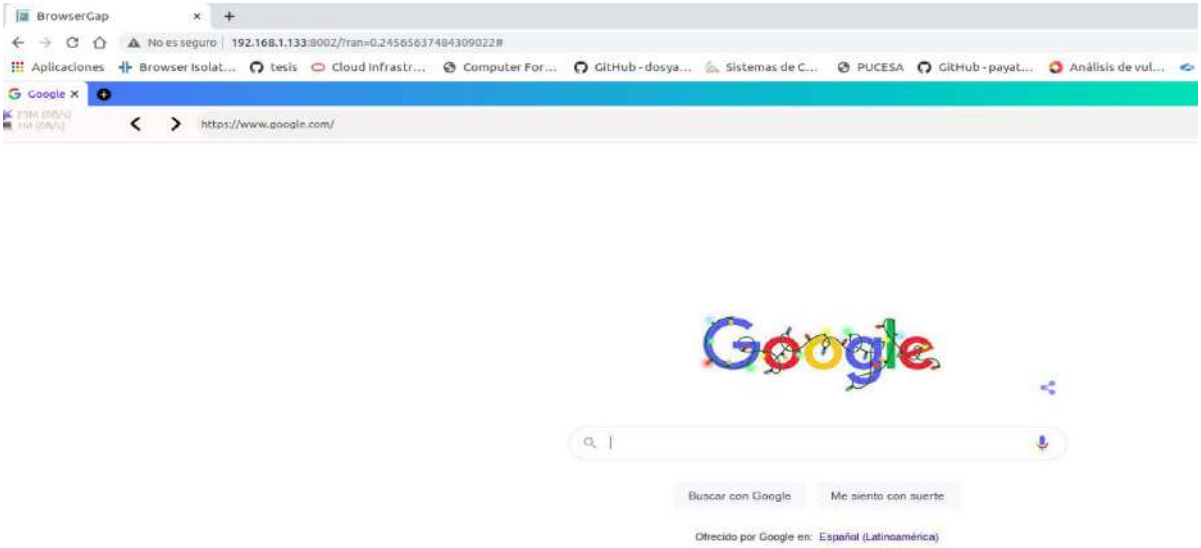
*Incremento 3:* Después de un análisis y pruebas individuales se escoge a Browsergap como api de isolación base, mismo que en historias siguientes se modificará para adaptarlo a las necesidades del proyecto. Se despliega bajo docker y se lo implementa en el servidor de pruebas. Sé considera que se modificará ampliamente la herramienta,

se realiza un *fork* del proyecto oficial hacia nuestro repositorio propio, mismo que se considerará como el oficial para esta implementación.

*Ilustración 8: Servicio web-isolation*


Fuente: elaboración propia

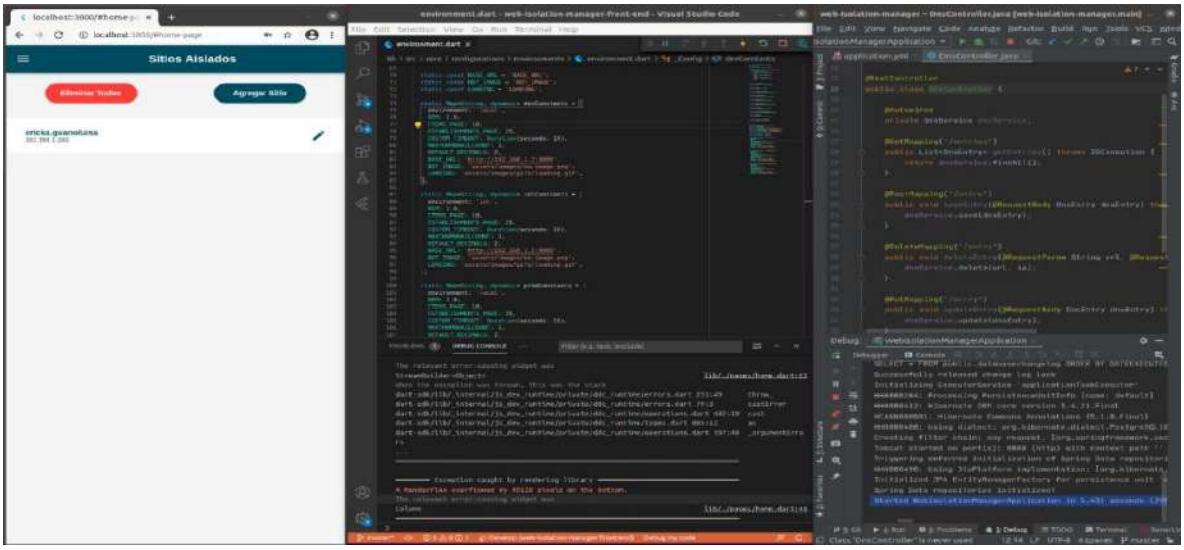
*Ilustración 9: Ejecución Web Isolation*




Fuente: elaboración propia

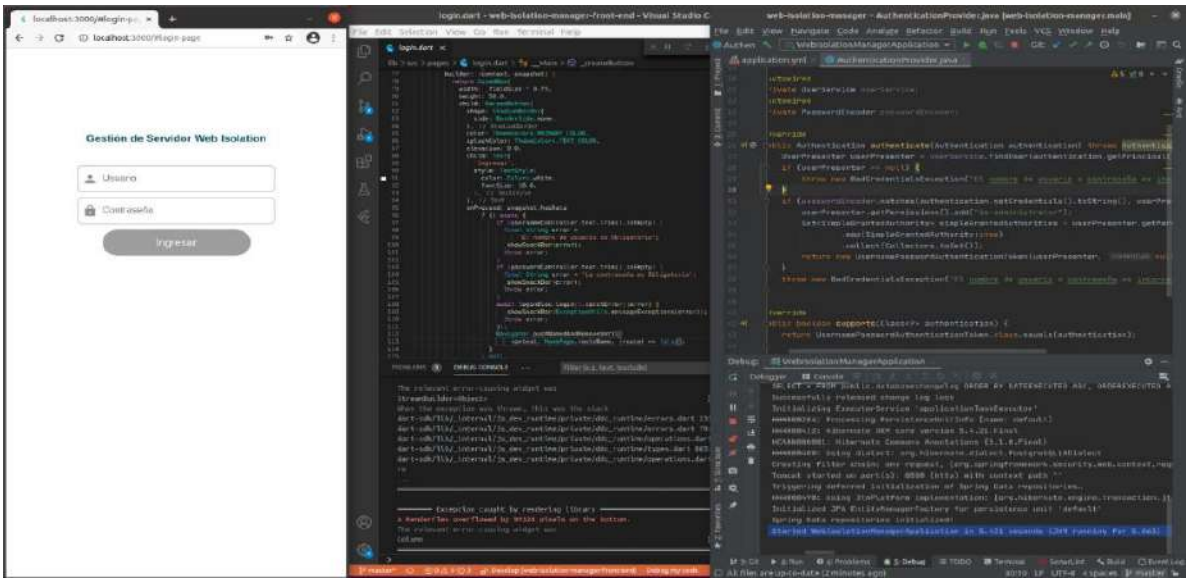
*Incremento 4:* Se considera herramientas modernas y lenguajes robustos y con mantenimiento, se crea un Rest Api para el *back end* con java y Spring Boot, mismo que gestionará las configuraciones de dnsmasq a través del volumen que compartan sobre el archivo de configuración, el *front end* se lo realiza en flutter para aprovechar las ventajas del desarrollo multiplataforma, en este caso se compiló para la web pero se compilaría tanto para android e ios el mismo código fuente, por eso el diseño sigue las tendencias para adaptarse a estos dispositivos. Para la seguridad y autenticación se utiliza Spring *Security* y tokens jwt.

*Ilustración 10: Gestión de sitios a aislar*




Fuente: elaboración propia

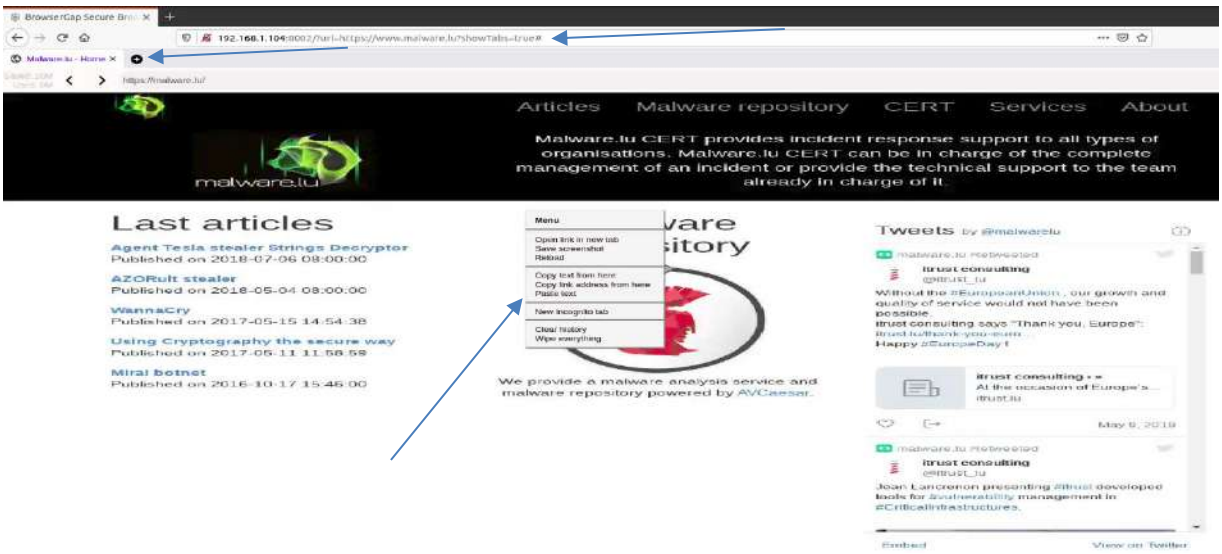
*Ilustración 11: Gestión de Usuarios y Roles*




Fuente: elaboración propia

*Incremento 5:* Como se requiere mantener privacidad en la navegación y dado que la herramienta solicita la renovación manual del *token* de identidad, se cambió este proceso para que sea automático, dado que, el sitio solicitado es el único que sería utilizado, por cuanto se remueve también la capacidad de crear nuevas pestañas al modificar los css y retirar las funciones javascript.

*Ilustración 12: URL con parámetro true*




Fuente: elaboración propia

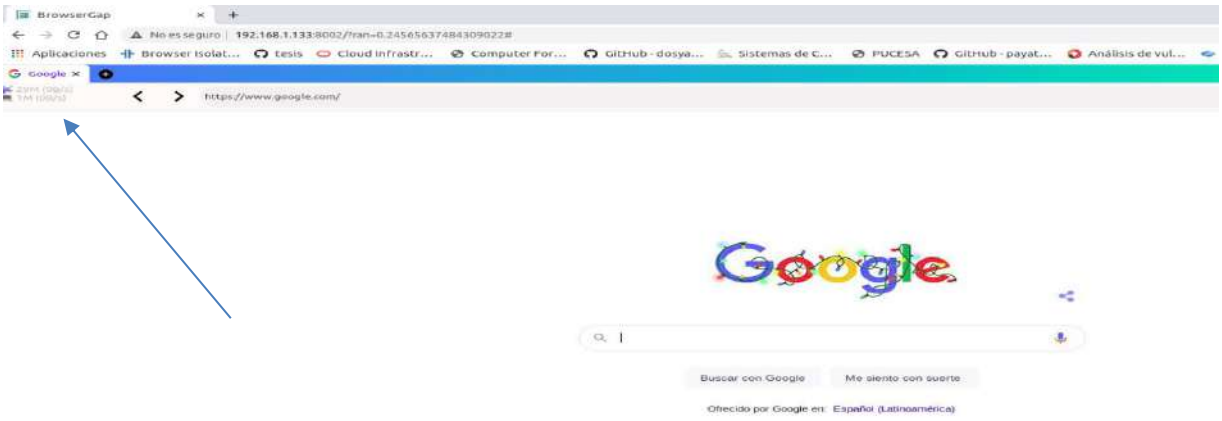
*Ilustración 13: URL con parámetro false*


Fuente: elaboración propia

Como se nota en las ilustraciones 12 y 13, el parámetro showTabs es el que activa o desactiva el modo restringido para el usuario, lo que permite así a los técnicos un mayor control sobre la herramienta sea para pruebas o desarrollo, esta configuración se cambiaría en el archivo nginx/conf.d/redirect.conf.

*Incremento 6:* Las estadísticas se implementan con el uso del patrón de desarrollo *provider* para poder mantener datos en tiempo real por cliente en tiempo de ejecución y poder visualizar los mismos sin consultas permanentes individuales, se colocan en un lugar visible y no utilizado, se escoge un color con gran opacidad para no causar molestias en los usuarios.

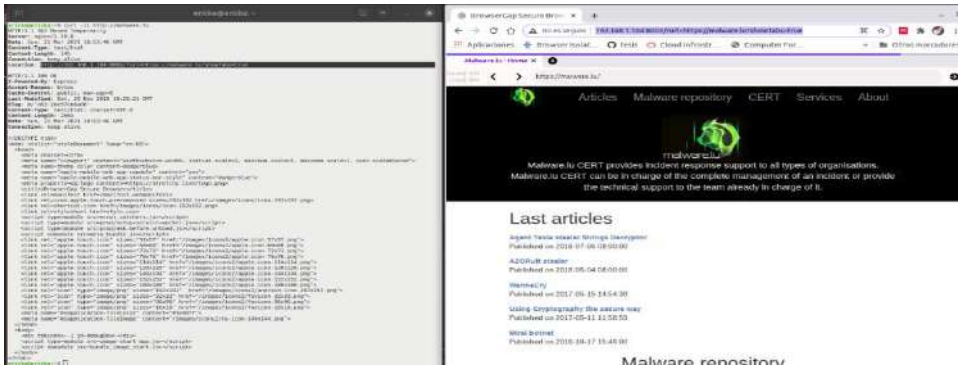
*Ilustración 14: Estadísticas de datos*




Fuente: elaboración propia

Incremento 7: Se utiliza la configuración del reverse proxy para enviar la url solicitada como parámetro al api y que este la resuelva automáticamente. Los sitios http como malware.lu son resueltos exitosamente e isolados por el api, como lo muestra la ilustración 15.

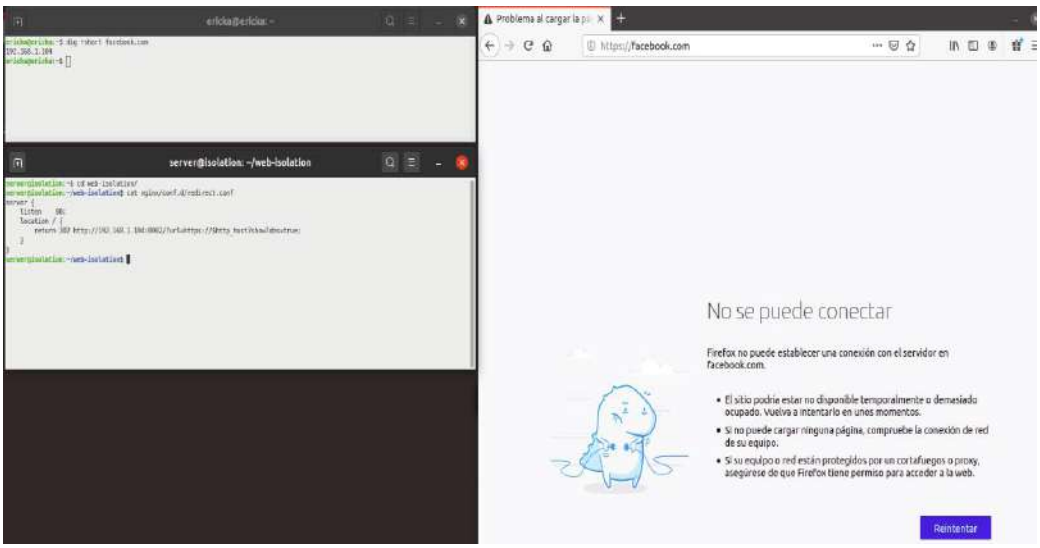
*Ilustración 15: Prueba reverse proxy http*




Fuente: elaboración propia

Se prueba con una red social en este caso facebook.com misma que es redireccionada a https por el navegador, por lo cual se nota el reverse proxy no está escucha https.

*Ilustración 16: Prueba reverse-proxy https*




Fuente: elaboración propia

Se implementa esta configuración, al ser un entorno local se utiliza certificados auto firmados con mkcert.

*Ilustración 17: Instalación certificados mkcert*


Fuente: elaboración propia

Posteriormente se actualiza el servidor nginx para acceder https, con lo cual se logra la redirección.

*Ilustración 18: Configuración nginx*

```
server@isolation:~/web-isolation$ cat nginx/conf.d/redirect.conf
server {
    listen       80;
    location / {
        return 302 http://192.168.1.104:8002/?url=https://$http_host?showTabs=true;
    }
}
server {
    listen 443 ssl;

    #server_name localhost;

    ssl_certificate /etc/ssl/private/localhost.pem;
    ssl_certificate_key /etc/ssl/private/localhost-key.pem;

    location / {
        #proxy_set_header X-Forwarded-Proto $scheme;
        return 302 http://$remote_addr:8002/?url=https://$http_host?showTabs=true;
    }
}
server@isolation:~/web-isolation$
```

Fuente: elaboración propia

*Ilustración 19: Visualización de certificados*

```
server@isolation:~/web-isolation$ ls certs/
localhost-key.pem  localhost.pem
server@isolation:~/web-isolation$
```

Fuente: elaboración propia

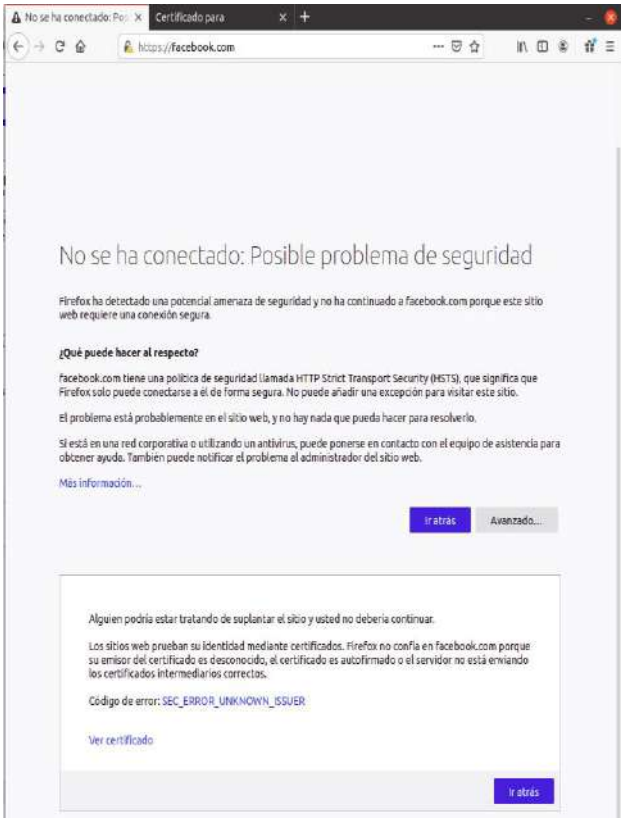
*Ilustración 20: Visualización de certificados*

```
    volumes:
      - ./nginx:/etc/nginx
      - ./certs:/etc/ssl/private
server@isolation:~/web-isolation$
```

Fuente: elaboración propia

Pero el navegador lo detecta como suplantación de identidad lo que impide la navegación a causa de incorcondacia en el emisor del certificado (nuestro vs original).

*Ilustración 21: Navegador indica Suplantación de identidad*




Fuente: elaboración propia

Incremento 8: A causa de las complicaciones anteriores, durante el review de fecha sptint 7 se decide desarrollar este nuevo sprint enfocado en intentar resolver el

problema de suplantación por cuanto se genera una alternativa software propia encargada de la redirección escrita en dart, la misma que se encuentra en el repositorio https://gitlab.com/tesis8/web-isolation-redirecter

*Ilustración 22: Redirección en dart*

main.dart 714 Bytes Edit Web IDE

```dart
import 'dart:html';
import 'package:url_launcher/url_launcher.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    _doRedirect();
    return Container();
  }

  Future _doRedirect() async {
    var url = window.location.href;
    if (await canLaunch(
        'http://localhost:8002/?url=https://www.facebook.com?showTabs=true')) {
      await launch(
          'http://localhost:8002/?url=https://www.facebook.com?showTabs=true');
      //window.close();
      print(window.closed);
    } else {
      throw 'Could not launch';
    }
  }
}
```

Fuente: elaboración propia

Se utiliza esta imagen en lugar del reverse proxy y se lo integra en el entorno, como se visualiza a continuación.

*Ilustración 23: Docker-compose actualizo con imagen de redirect realizado*

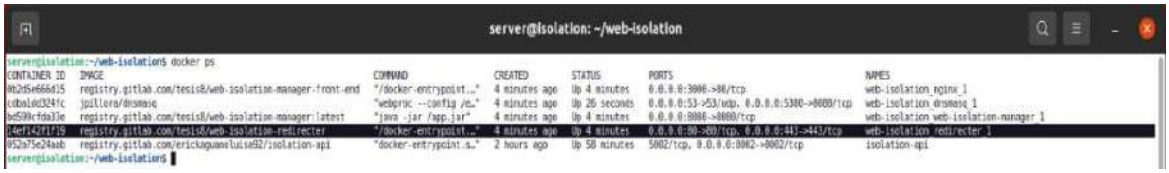
docker-compose.yml 1.1 KB Edit Web IDE

```yaml
version: '3'

services:
  dnsmasq:
    ports:
      - '53:53/udp'
      - '5380:8080'
    volumes:
      - ./dnsmasq.conf:/etc/dnsmasq.conf
    logging:
      options:
        max-size: 100m
    environment:
      - HTTP_USER=foo
      - HTTP_PASS=bar
    restart: unless-stopped
    image: jpillora/dnsmasq
  web-isolation-manager:
    image: registry.gitlab.com/tesis8/web-isolation-manager:latest
    expose:
      - '8080'
    ports:
      - '${BACKEND_PORT}:8080'
    volumes:
      - ./dnsmasq.conf:/opt/dnsmasq.conf
    environment:
      - TZ=America/Guayaquil
    restart: unless-stopped
  nginx:
    image: registry.gitlab.com/tesis8/web-isolation-manager-front-end
    expose:
      - "80"
    ports:
      - '3000:80'
    depends_on:
      - web-isolation-manager
    environment:
      - BACKEND_IP=${BACKEND_IP}
      - BACKEND_PORT=${BACKEND_PORT}
      - BACKEND_CONTEXT=${BACKEND_CONTEXT}
    restart: unless-stopped
  redirecter:
    image: registry.gitlab.com/tesis8/web-isolation-redirecter
    ports:
      - 80:80
      - 443:443
    volumes:
      - ./nginx:/etc/nginx
      - ./certs:/etc/ssl/private
```

Fuente: elaboración propia

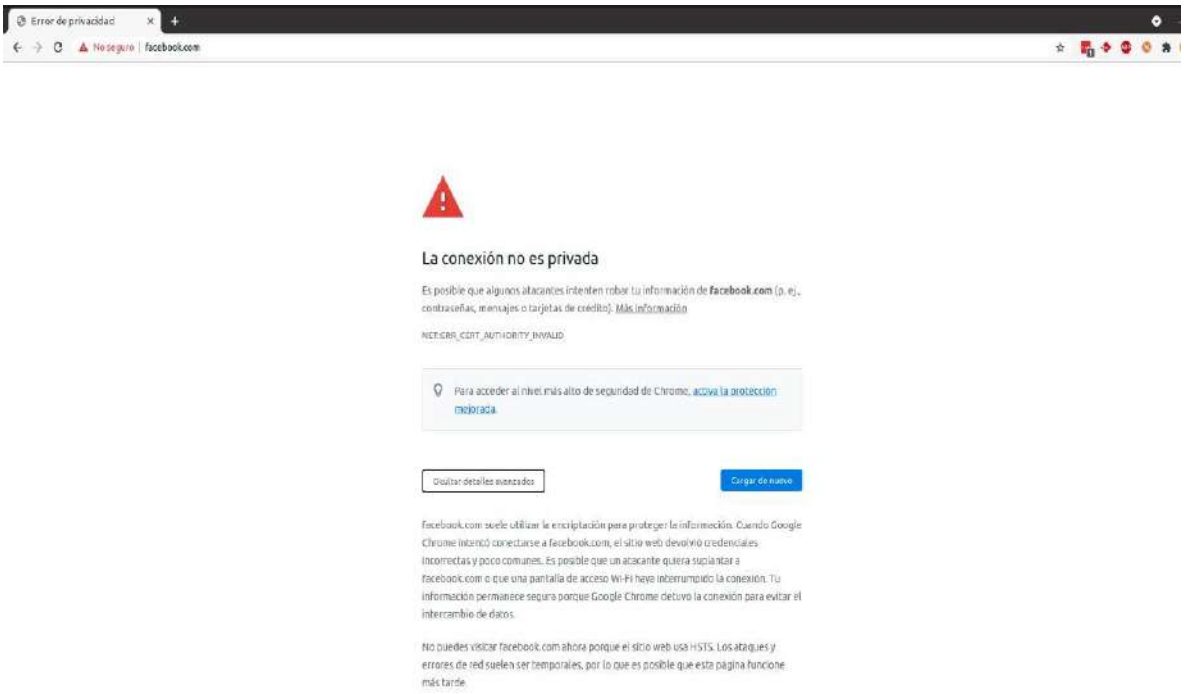
*Ilustración 24: Docker-ps*




Fuente: elaboración propia

Se obtuvo el mismo resultado de suplantación de identidad.

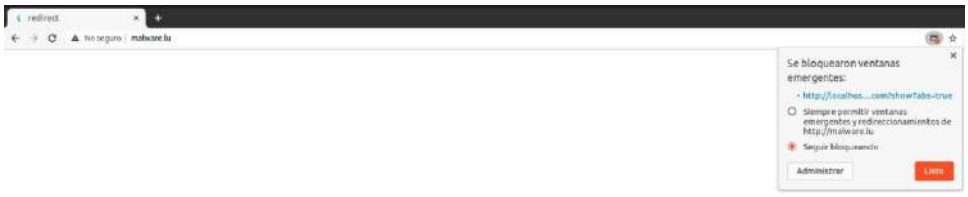
*Ilustración 25: Navegador indica Suplantación de identidad*




Fuente: elaboración propia

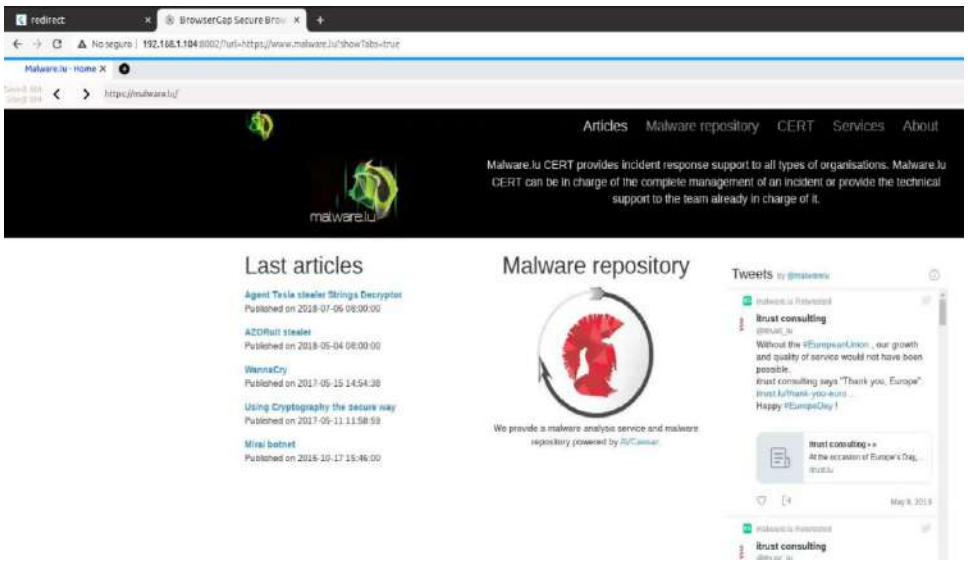
Además, se evidencia una menor velocidad en la navegación, así como la necesidad de abrir nuevas pestañas del navegador por cuanto se retoma la solución de la iteración 7.

*Ilustración 26: Redirect en funcionamiento*




Fuente: elaboración propia

*Ilustración 27: Con Redirect se abre una nueva pestaña*

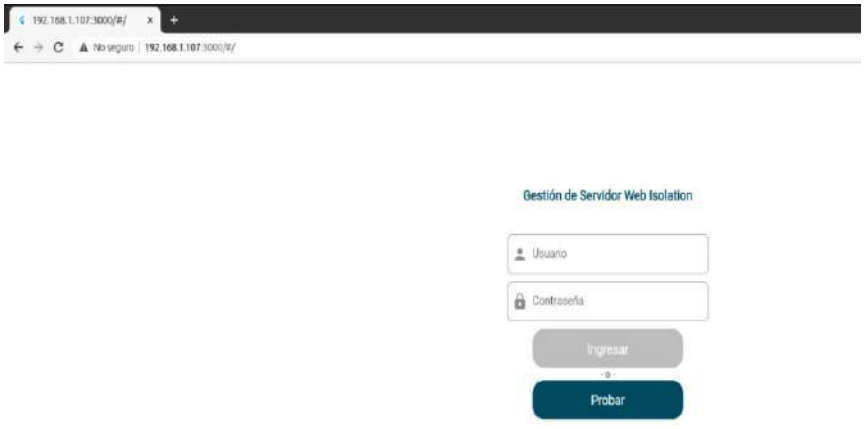



Fuente: elaboración propia

*Incremento 9:* Durante la *restrospective* del incremento anterior, se define como necesaria una manera de islolar sitios https, lo cual genera como resultado el sitio isolate-me, que se accede desde la opción probar que se agrega en el login, el mismo ofrece una caja de texto en donde se escribirá la dirección a visitar de manera segura y un botón Ir que genera la navegación.

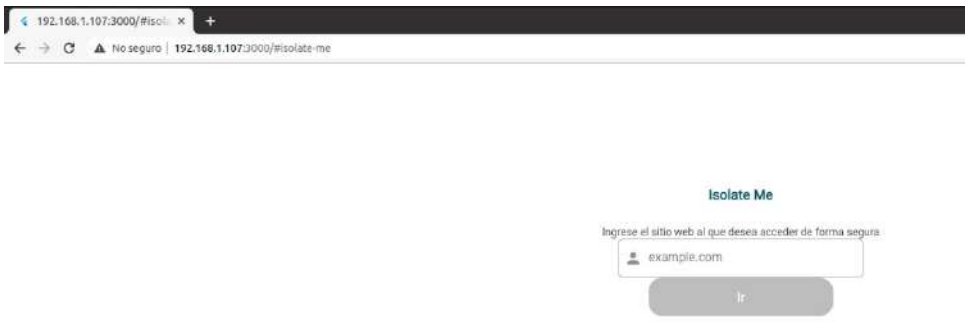
Se decide dejarla implementada como una solución complementaria, dado que, no es totalmente transparente para el usuario final, pero es funcional en caso de requerirse.

*Ilustración 28: Botón probar*




Fuente: elaboración propia

*Ilustración 29: Pagina isoalte-me*




Fuente: elaboración propia

El código generado en los incrementos del desarrollo de la presente propuesta se encuentra en los proyectos de gitlab en la rama master.

- https://gitlab.com/erickaguanoluisa92/isolation-api
- https://gitlab.com/tesis8/web-isolation-manager-front-end
- https://gitlab.com/tesis8/web-isolation-manager
- https://gitlab.com/tesis8/web-isolation-redirecter

#### 2.1.2.5. Sprint Review

Los incrementos obtenidos durante todo el Sprint son presentados durante esta reunión al *product owner* y demás participantes, así como su funcionamiento y condiciones, se trata de una socialización y no sería vista como un *check list*, la duración tiene un límite de tiempo de máximo cuatro horas para un Sprint de un mes. Para el desarrollo del proyecto dado que se definieron Sprint más cortos, el evento se lo estableció en máximo 2 horas de duración.

#### 2.1.2.6. Sprint Retrospective

Posterior al *review*, se utiliza esta reunión para poder evolucionar como equipo, se discute principalmente, lo que se hizo bien durante el sprint terminado, lo que se mejoraría y como hacerlo en los siguientes Sprint. Scrum establece un tiempo limitado a máximo tres horas para un sprint de un mes. En el desarrollo del proyecto, dado que,

se definieron Sprint más cortos por lo que este evento se lo estableció en máximo 1 hora con 30 minutos de duración.

*Tabla 6: Sprint Retrospective*

| Sprint | Fecha Terminación del Sprint | ***Retrospective*** |
|---|---|---|
| 1 | 16/09/2020 | ¿Qué se hizo bien?<br>Elegir herramientas de software libre en constante actualización.<br>Utilización de herramientas web, permitió el trabajo remoto en diferentes equipos<br>¿Qué se mejoraría?<br>Considerar los recursos disponibles en el equipo de desarrollo para garantizar que el ambiente de pruebas sea ejecutado localmente.<br>¿Cómo mejorar para el próximo Sprint?<br>Configurar las herramientas para que se ejecuten con requisitos mínimos. |
| 2 | 02/10/2020 | ¿Qué se hizo bien?<br>Utilización de docker para escalabilidad y fácil migración.<br>Cumplimiento del sprint goal en los tiempos requeridos.<br>¿Qué se mejoraría?<br>Utilizar volúmenes para la información de los contenedores<br>¿Cómo mejorar para el próximo Sprint?<br>Configurar las herramientas para que se ejecuten con requisitos mínimos. |
| 3 | 18/10/2020 | ¿Qué se hizo bien?<br>Utilización proyectos públicos de github.<br>Utilización de docker como plataforma de despliegue.<br>Cumplimiento del sprin *goa*l en los tiempos requeridos.<br>¿Qué se mejoraría?<br>Reducir las posibles maneras de evadir el control de la herramienta.<br>La navegación aislada será transparente para el usuario.<br>¿Cómo mejorar para el próximo Sprint?<br>Bloquear el tráfico del puerto 53 para prevenir el uso de un servidor dns externo no autorizado.<br>Al tomar la historia de usuario 6, modificar el docker-compose, para agregar un volumen hacia el archivo dnsmaq.conf |
| 4 | 03/11/2020 | ¿Qué se hizo bien?<br>Uso de spring *security* para la gestión de credenciales.<br>Utilizar un hash bcrypt para asegurar las contraseñas en la base de datos.<br>¿Qué se mejoraría?<br>Diferenciar los mensajes error entre usuarios inexistentes y contraseña incorrecta.<br>¿Cómo mejorar para el próximo Sprint?<br>Agregar estilos a la interfaz de usuario. |

| | | |
|---|---|---|
| 5 | 19/11/2020 | ¿Qué se hizo bien?<br>*Forkear* el repositorio oficial a uno privado.<br>¿Qué se mejoraría?<br>En la redacción de la historia agregar las funcionalidades mínimas requeridas para dar por terminado una historia.<br>¿Cómo mejorar para el próximo Sprint?<br>Mejorar la definición de hecho dado que han existido inconvenientes para dar por terminado la historia. |
| 6 | 05/12/2020 | ¿Qué se hizo bien?<br>Mejorar la definición de hecho.<br>¿Qué se mejoraría?<br>Mejorar el *responsive* para dispositivos móviles.<br>¿Cómo mejorar para el próximo Sprint?<br>Agregar una tarea en el sprint 7 para la mejora de esta característica. |
| 7 | 21/12/2020 | ¿Qué se hizo bien?<br>Utilizar tecnologías populares por su amplia documentación<br>¿Qué se mejoraría?<br>Horarios de las reuniones, dado que ha este punto del proyecto las revisiones son más extensas.<br>¿Cómo mejorar para el próximo Sprint?<br>Realizar las reuniones en fines de semana antes del mediodía. |
| 8 | 26/03/2020 | ¿Qué se hizo bien?<br>Reutilizar tecnologías conocidas.<br>Realizar pruebas de concepto durante el desarrollo para poder descartar soluciones antes de terminarlas.<br>¿Cómo mejorar para el próximo Sprint?<br>En caso de funcionalidades desconocidas, agregar una reunión a la mitad del sprint. |
| 9 | 19/03/2020 | ¿Qué se hizo bien?<br>Comunicación para definición de alternativas<br>¿Qué se mejoraría?<br>Culminación de proyecto no aplica.<br>¿Cómo mejorar para el próximo Sprint?<br>Culminación de proyecto no aplica. |

Fuente: elaboración propia

## 2.2. Metodología de la Investigación

La presente propuesta tiene un enfoque experimental, por cuanto se requiere establecer comparaciones del antes y después de la implementación de la solución de Web *Isolation*, es así que basados en este requerimiento se considera como metodología de investigación a la experimental.

A continuación, se describen detalladamente los procesos realizados en cada una de las fases planteadas por esta metodología.

### 2.2.1. Planteamiento del problema

El problema científico de la presente investigación se define: ¿Cómo disminuir el costo de la implementación de una herramienta de *Web Isolation* para el aseguramiento de la navegación de los equipos clientes?

### 2.2.2. Planteamiento de la hipótesis

La implementación de una herramienta accesible de *Web Isolation*, permitirá disminuir los costos del aseguramiento de los equipos clientes.

### 2.2.3. Definición de variables

Variables Dependientes:

- Vulnerabilidades mitigadas
- Costo de implementación
- Costo de mantenimiento
- Velocidad de navegación

Variable Independiente: Implementación de la solución Web Isolation

### 2.2.4. Operacionalización de variables

*Tabla 7: Operacionalidad de variables dependientes*

| Tipo | Variable | Descripción | Indicador | Técnicas | Instrumentos |
|---|---|---|---|---|---|
| **Dependiente** | Vulnerabilidades mitigadas | Número entero de incidentes detectados con y sin el uso de la herramienta. | Cantidad de incidentes | Observación directa | Registros |
| | Costo de implementación | Valor monetario en dólares americanos que representan el presupuesto requerido para la primera puesta en marcha de la herramienta. | Cantidad en dólares | Observación directa | Registros |
| | Costo de mantenimiento | Valor monetario en dólares americanos que representan el presupuesto requerido para conservar la herramienta en funcionamiento a través del tiempo. | Cantidad en dólares | Observación directa | Registros |

| | | | | | |
|---|---|---|---|---|---|
| | Velocidad de navegación | Lapso de tiempo que representa la obtención del sitio web al usuario final. Se mide desde el momento que se realiza la petición hasta que todos los recursos se encuentran en el navegador | Tiempo en milisegundos | Observación directa | Software |

Fuente: elaboración propia

*Tabla 8: Operacionalidad de variables independientes*

| Tipo | Variable | Descripción | Método | Herramientas |
|---|---|---|---|---|
| **Independiente** | Implementación de la solución Web *Isolation* | Diseño de una herramienta que permita aislar sitios web. | SCRUM | -Entorno integrado de desarrollo<br>-Docker |

Fuente: elaboración propia

### 2.2.1. Procedimiento y recolección de datos

Se utilizó la técnica de observación y búsqueda de información bibliográfica en sitios web sobre la solución propuesta, se consideró el idioma inglés y español: *Web Isolation, Browser Isolation,* Navegador Aislado, Asilamiento de Navegador.

Además, se aplicaron técnicas de experimentación, para el análisis de resultados, esto para los experimentos del antes y después, en el antes se realizó las pruebas en los equipos clientes sin el uso de la herramienta de *Web Isolation* implementada, paralelamente se midió la velocidad y cantidad de amenazas mitigadas; para el después se utilizó los mismos equipos clientes, pero con el uso de la solución planteada. Para la verificación de la hipótesis se utilizó el software R como herramienta automatizada de análisis estadístico como se evidencia en el capítulo III: Análisis de los resultados de la Investigación.

## CAPÍTULO III. ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN

Se obtiene una aplicación que funciona bajo los requerimientos definidos inicialmente, así como las capacidades adicionales generadas durante las iteraciones, mismas que se ven limitadas únicamente por la funcionalidad propia de los navegadores actuales que pre procesan la navegación antes de ejecutarla.

El caso particular de la red social facebook.com, misma que en el navegador chrome como en firefox no permite su isolación de manera transparente, pese a que una petición http pura si atraviesa el entorno de isolación, proceso que se detallada continuación:

1. Se revisa el dns que resuelve facebook.com, misma que es correcta.

*Ilustración 30: Resolución dns a Facebook.com*


Fuente: elaboración propia

2. Se realiza una petición http misma que responde un código de estado http 302(moved).

*Ilustración 31: Petición http*


Fuente: elaboración propia

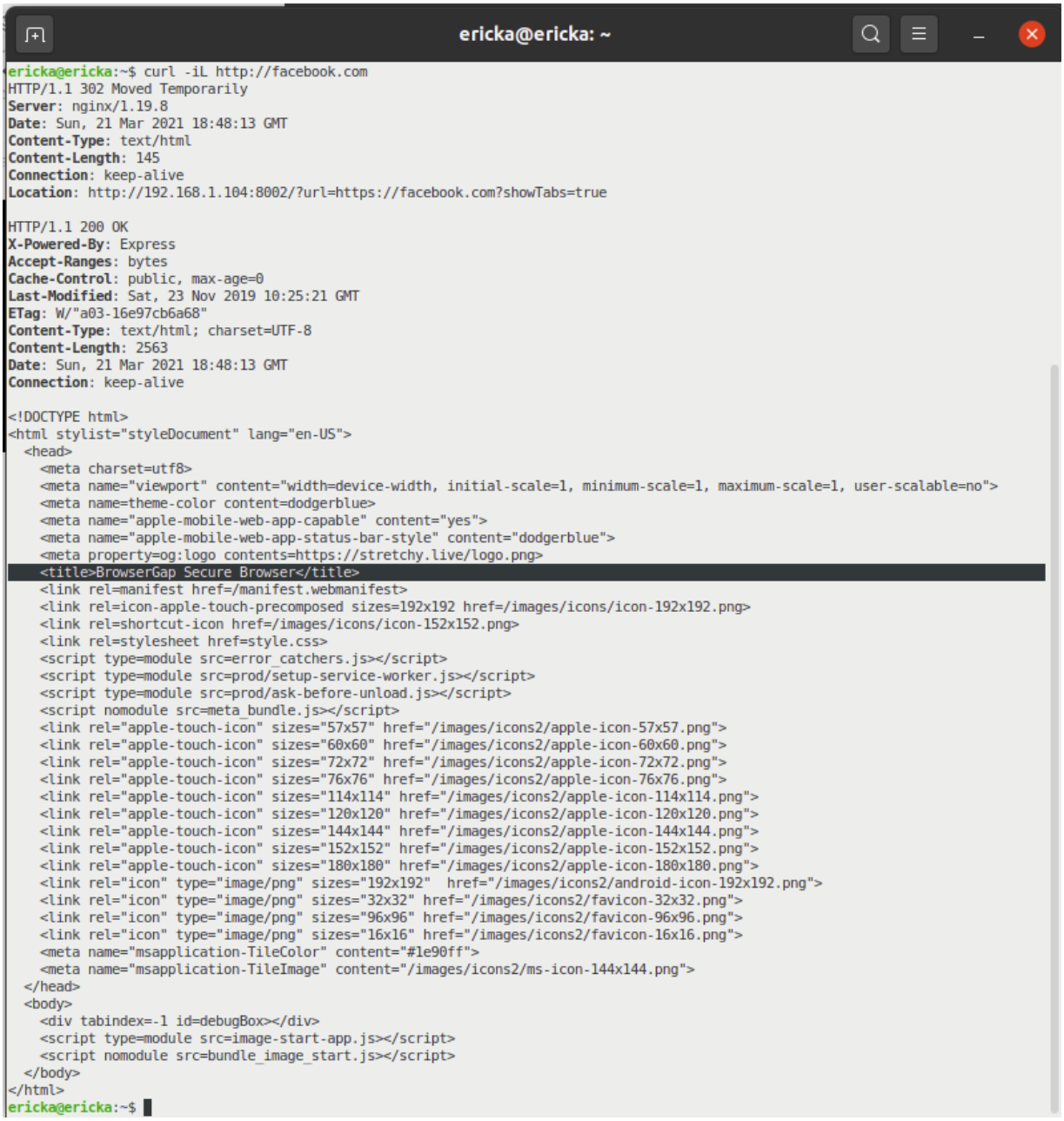
3. Para verificar a donde se está realizado la redirección se agrega el parámetro i, el cual revela una correcta redirección al servidor de isolación.

*Ilustración 32: Petición http con parámetro -i*


Fuente: elaboración propia

4. para resolver la redirección se agrega el parámetro –L lo cual devuelve el html del servidor de isolación.

*Ilustración 33: Petición http con parámetro -iL*

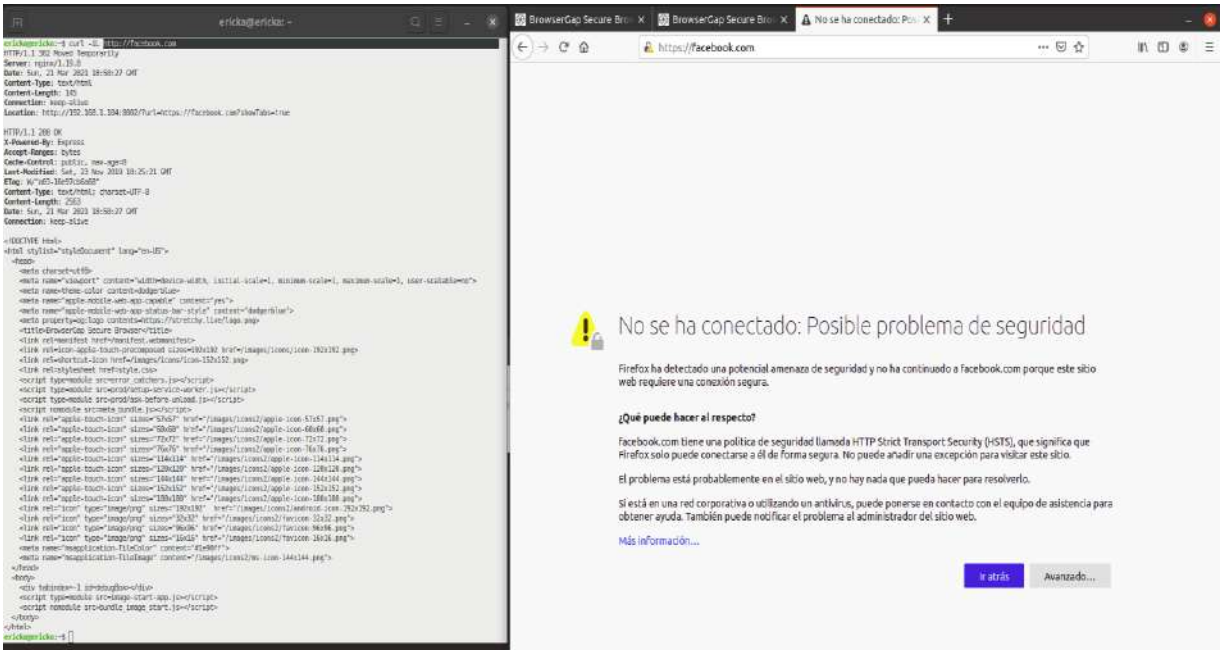



Fuente: elaboración propia

5. En el mismo entorno se prueba desde el navegador y se obtiene un error y una redirección causada por el mismo, lo cual muestra un pre procesamiento o telemetría a nuestra navegación antes de su conexión a internet.

Este problema relacionado a hsts se intenta resolver en el incremento 8 cuyos resultados fueron expuestos en el capítulo 2.

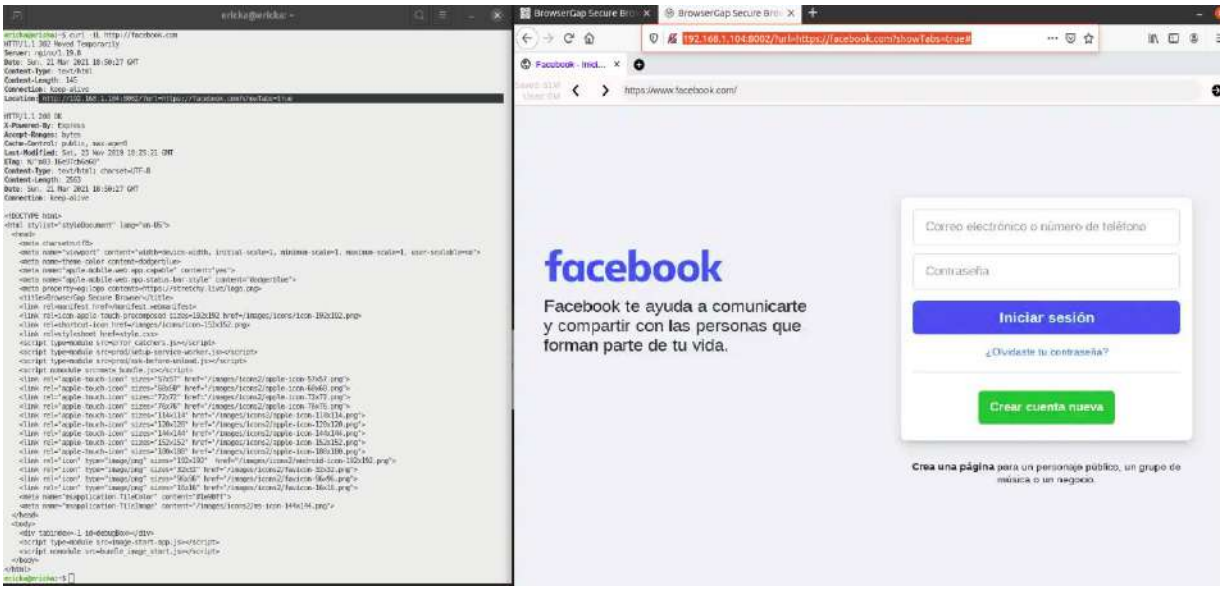
*Ilustración 34: Suplantación de identidad por certificado no coincidente*




Fuente: elaboración propia

6. Se prueba manualmente la dirección resultante de la petición, misma que funciona correctamente, por cuanto a limitante se encuentra en el navegador.

*Ilustración 35: Prueba exitosa de la redirección*




Fuente: elaboración propia

## 3.1. Pruebas de la Investigación

Se realizaron las pruebas en un entorno controlado de 20 equipos clientes, en el cual se procede con el monitoreo continuo para medir el impacto de la herramienta, en función de:

### 3.1.1. Ataques realizados vs ataques mitigados

Para esta prueba se establece un listado con los sitios con vulnerabilidades conocidas, los cuales, se van a acceder, se espera que la implementación de la herramienta limite el acceso de software malicioso a los equipos clientes.

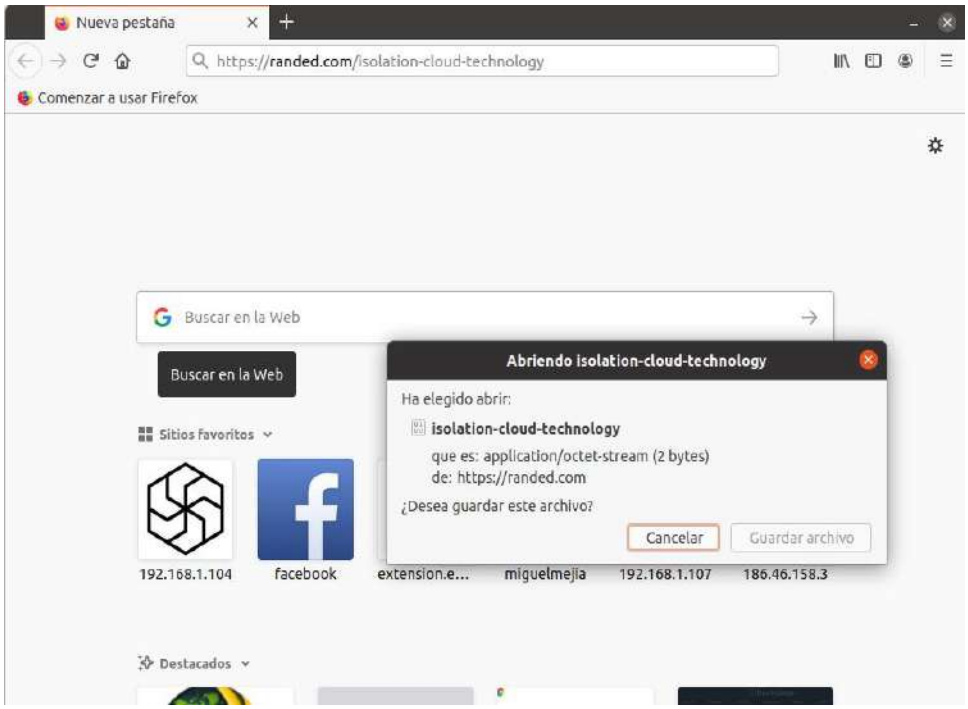
Para la medición se seleccionan los siguientes sitios:

- Sitio 1: https://randed.com/isolation-cloud-technology/
- Sitio 2: https://malware.erickaguanoluisa.ml/
- Sitio 3: http://extension.erickaguanoluisa.ml/

Los cuales entregan 1 malware cada uno, lo que da un total de 3.

Caso 1, el sitio es visitado por los equipos sin la herramienta y como se visualiza en la Ilustración 36, descarga un archivo bin peligroso para el sistema operativo.

*Ilustración 36: Sitio 1 sin el uso de la herramienta*




Fuente: elaboración propia

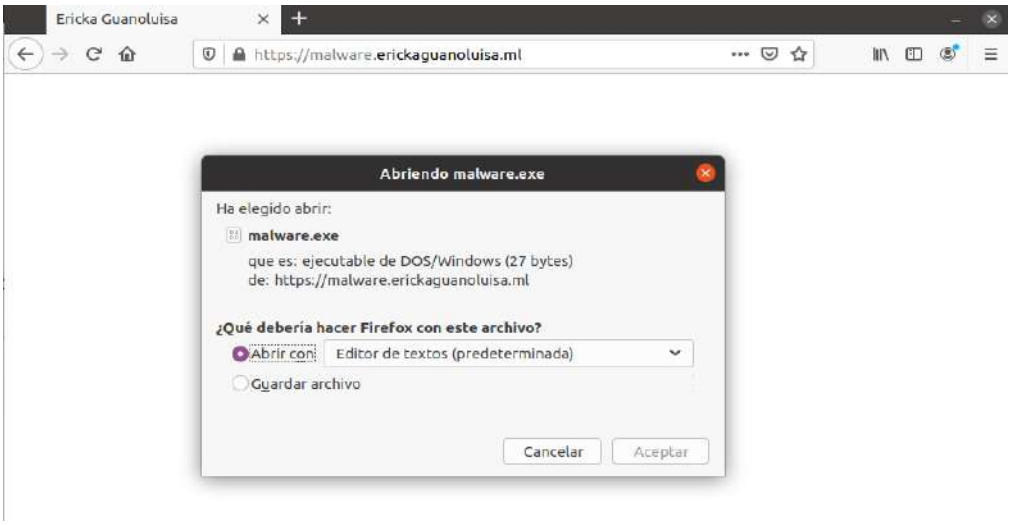
A continuación, se vuelven a construir los equipos clientes y se implementa la solución, se procede a visitar nuevamente el sitio comprometido, como lo muestra la ilustración 37, el equipo cliente no permite la descarga del archivo malicioso e interrumpe la navegación al mismo.

*Ilustración 37: Sitio 1 con el uso de la herramienta*


Fuente: elaboración propia

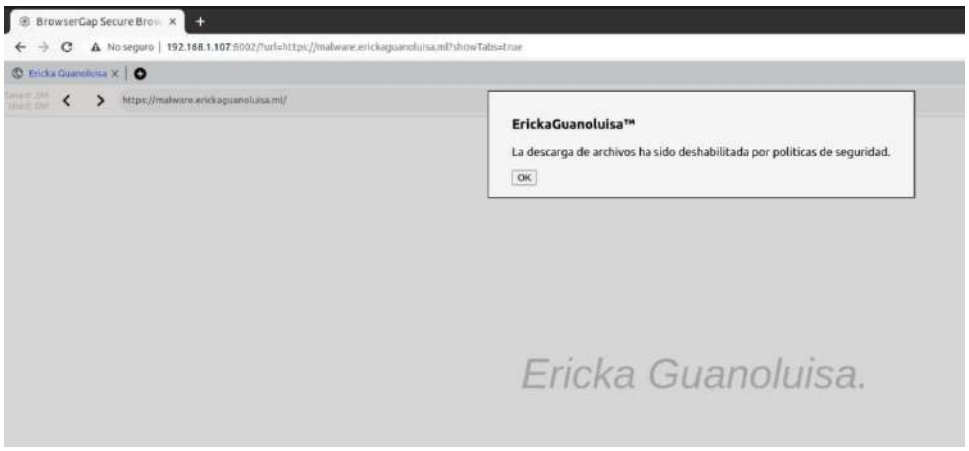
En el caso 2, al tratarse de un sitio nuevo, que no se encuentra reportado, el adjunto intenta descargarse, por cuanto se implementa un bloqueo adicional por software para este tipo de amenazas, la ilustración 38 muestra la alerta emitida con la herramienta.

*Ilustración 38: Sitio 2 sin el uso de la herramienta*




Fuente: elaboración propia

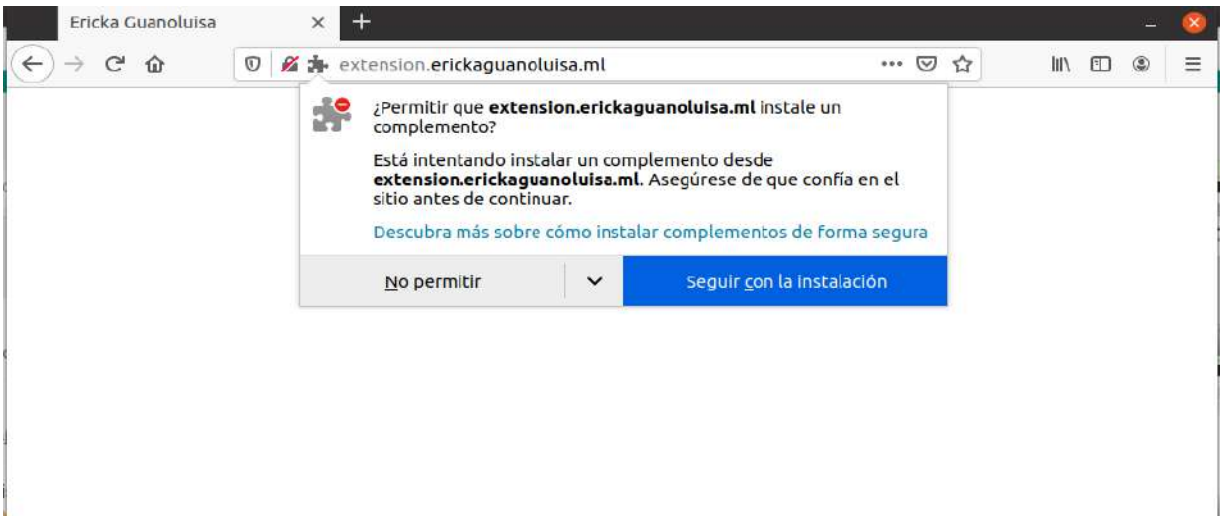
*Ilustración 39: Sitio 2 con el uso de la herramienta*




Fuente: elaboración propia

En el caso 3, el sitio descarga una extensión maliciosa en el navegador, la cual se anexa al mismo para funcionar como se muestra en la ilustración 40, se procede de la misma manera que en el caso anterior y se evidencia que, sin la herramienta, el navegador del cliente final resulta afectado, mientras que, con el uso de la herramienta, la descarga e instalación del mismo no se realiza, lo que permite lograr así aislar al cliente de este tipo de amenazas, además, si el navegador llegase a verse comprometido, al actualizar la página web o volver a acceder al sitio, el api de isolación entregará un navegador completamente nuevo.

*Ilustración 40: Sitio 3 sin el uso de la herramienta*




Fuente: elaboración propia

*Ilustración 41: Sitio 3 con el uso de la herramienta*

Fuente: elaboración propia

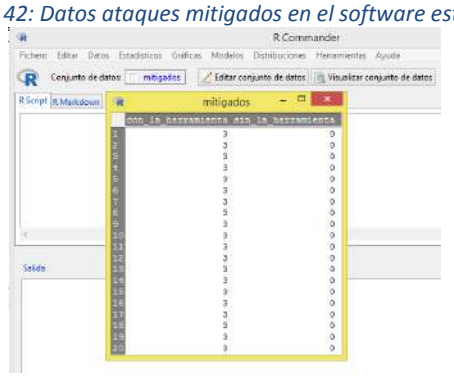
*Tabla 9: Medición variable dependiente Ataques mitigados*

| Equipo | Ataques Mitigados | |
|---|---|---|
| | **Sin la herramienta** | **Con la herramienta** |
| 1 | 0 | 3 |
| 2 | 0 | 3 |
| 3 | 0 | 3 |
| 4 | 0 | 3 |
| 5 | 0 | 3 |
| 6 | 0 | 3 |
| 7 | 0 | 3 |
| 8 | 0 | 3 |
| 9 | 0 | 3 |
| 10 | 0 | 3 |
| 11 | 0 | 3 |
| 12 | 0 | 3 |
| 13 | 0 | 3 |
| 14 | 0 | 3 |
| 15 | 0 | 3 |
| 16 | 0 | 3 |
| 17 | 0 | 3 |
| 18 | 0 | 3 |
| 19 | 0 | 3 |
| 20 | 0 | 3 |
| **Promedio** | **0** | **3** |

Fuente: elaboración propia

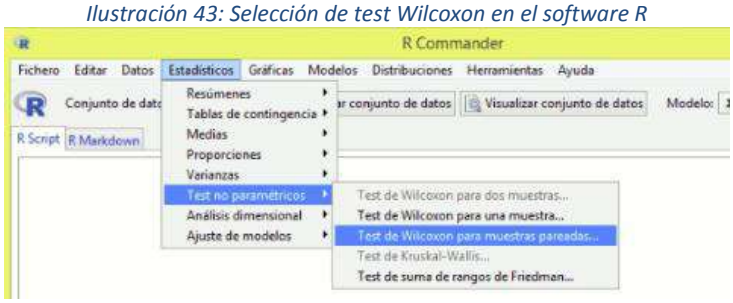
Dado que los valores son constantes se desestiman, por lo cual no tienen una distribución normal y para su análisis se aplica Wilcoxon, este test realizado en el software estadístico R, como se indica a continuación.

*Ilustración 42: Datos ataques mitigados en el software estadístico R*



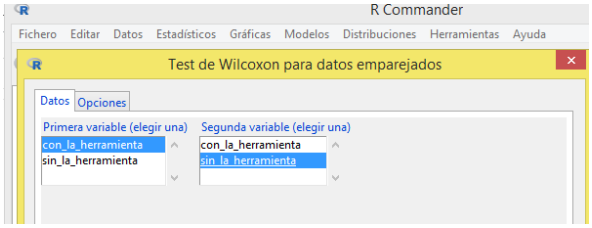

Fuente: elaboración propia

*Ilustración 43: Selección de test Wilcoxon en el software R*



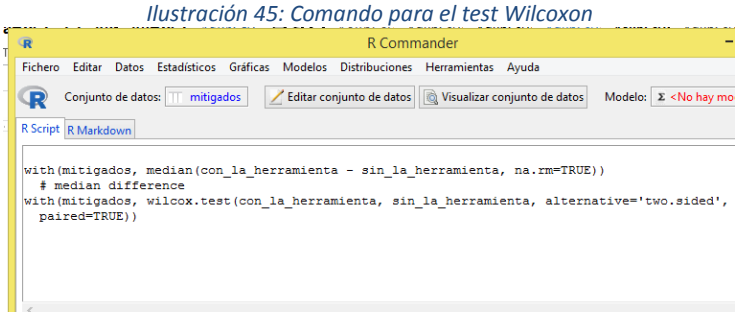

Fuente: elaboración propia

*Ilustración 44: Selección de variables para el test Wilcoxon*



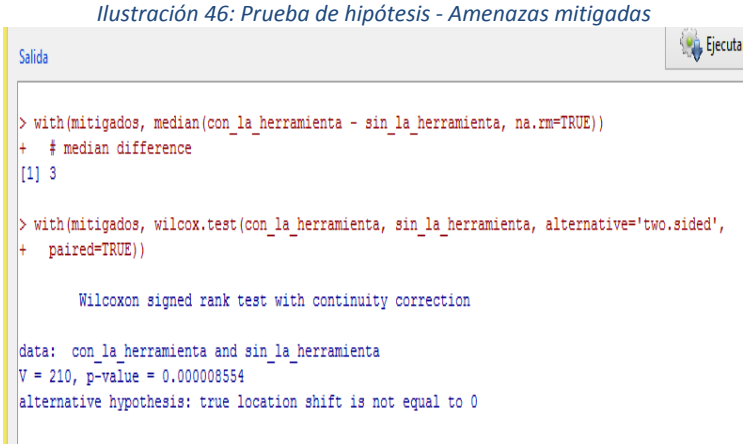

Fuente: elaboración propia

*Ilustración 45: Comando para el test Wilcoxon*




Fuente: elaboración propia

*Ilustración 46: Prueba de hipótesis - Amenazas mitigadas*




Fuente: elaboración propia

**Análisis**: Con la aceptación de un error igual a 0.05 equivalente al 95% de acierto, y con un p valor de 0.000008554 menor al nivel de significancia de 0.05, se concluye que con el uso de la herramienta permite mitigar la descarga de software malicioso en infraestructuras de red controladas, de los ejemplos planteados se logró neutralizar el 100% de los ataques como lo evidencia la Tabla 09.

### 3.1.2. Velocidad de navegación con la herramienta y sin la herramienta

Para el presente experimento y con el fin de obtener aleatoriedad, se decide tomar como sitio de pruebas al último visitado en el computador de desarrollo, mismo que es www.malware.lu, por cuanto se establece la prueba de velocidad en relación al mismo.

Como instrumento de medición se optará por la consola de chrome, misma que entregará la medida de tiempo en segundos en que un sitio logró ser totalmente presentado al cliente, se escoge este método, dado que, otros como cronómetros visuales o herramientas externas no son capaces de considerar procesos en segundo plano necesarios por los sitios para su funcionamiento.

Para mantener la uniformidad en el experimento, los equipos clientes se encenderán al momento de la prueba y el primer sitio que visiten será el seleccionado tanto con y sin la herramienta, resultados que se resumen en la Tabla 10.

*Tabla 10: Medición variable dependiente Velocidad de Navegación*

| Equipo | Velocidad de Navegación | |
|---|---|---|
| | **Sin la herramienta** | **Con la herramienta** |
| 1 | 4.06 s | 6.14 s |
| 2 | 3.41 s | 2.52 s |
| 3 | 6.87 s | 2.66 s |
| 4 | 4.03 s | 4.48 s |
| 5 | 3.95 s | 3.70 s |
| 6 | 3.48 s | 3.31 s |
| 7 | 3.33 s | 5.09 s |
| 8 | 3.07 s | 2.23 s |
| 9 | 4.11 s | 2.51 s |
| 10 | 4.99 s | 2.65 s |
| 11 | 3.18 s | 4.02 s |
| 12 | 2.14 s | 1.55 s |
| 13 | 2.59 s | 2.90 s |
| 14 | 4.38 s | 1.54 s |
| 15 | 2.04 s | 2.04 s |
| 16 | 2.52 s | 1.01 s |
| 17 | 2.33 s | 1.87 s |
| 18 | 2.03 s | 2.00 s |
| 19 | 2.38 s | 5.18 s |
| 20 | 1.67 s | 3.14 s |
| **Promedio** | **3.33 s** | **3.03 s** |

Fuente: elaboración propia

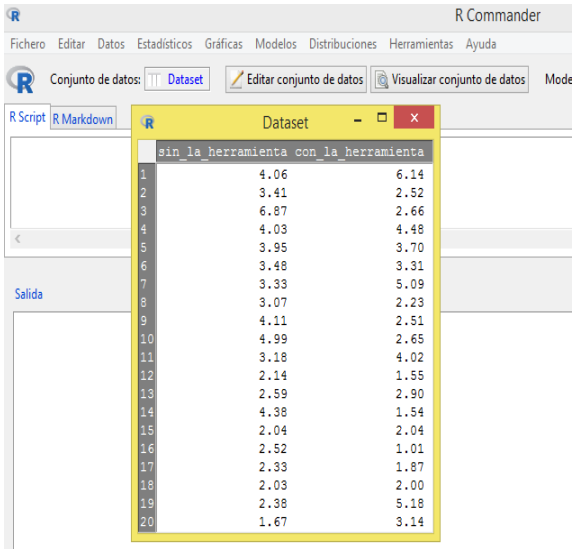
*Tabla 11: Pruebas de Normalidad*

| Variable | Parámetros | Significancia |
|---|---|---|
| **Velocidad de Navegación** | Sin la herramienta | 0.07302 |
| | Con la herramienta | 0.2544 |

Fuente: elaboración propia

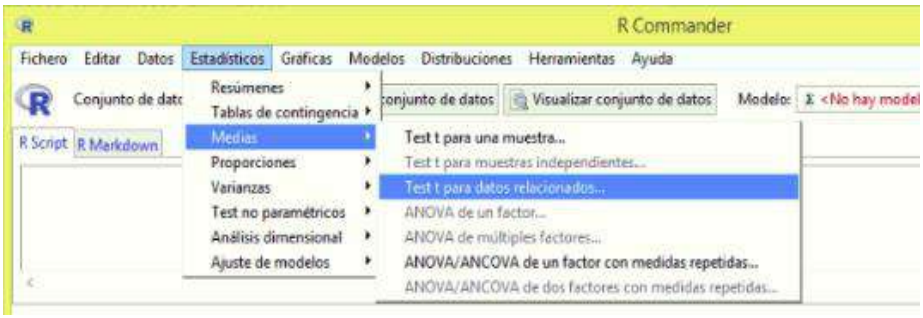
Los valores de significancia de la prueba de normalidad de Shapiro-Wilk aplicada a la variable velocidad de navegación, indican que son superiores al nivel de significancia planteada, por lo que se deduce que siguen una distribución normal, por tal motivo para su análisis se aplica t de Student, mediante el software R como se demuestra en las ilustraciones siguientes.

*Ilustración 47: Datos de velocidad de Navegación en el software estadístico R*



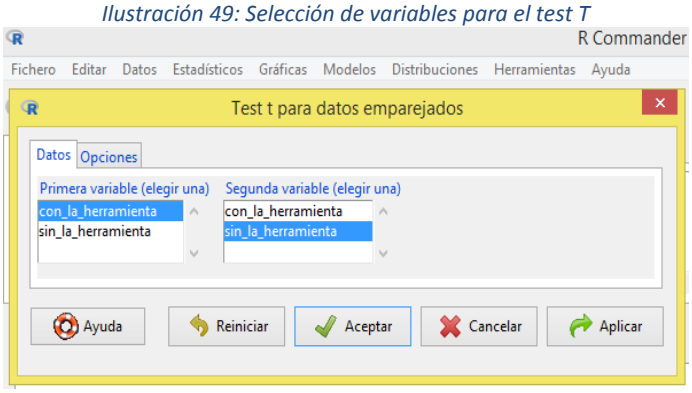

Fuente: elaboración propia

*Ilustración 48: Selección de test T*



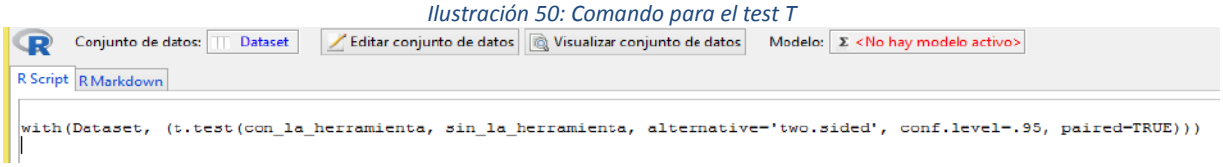

Fuente: elaboración propia

*Ilustración 49: Selección de variables para el test T*



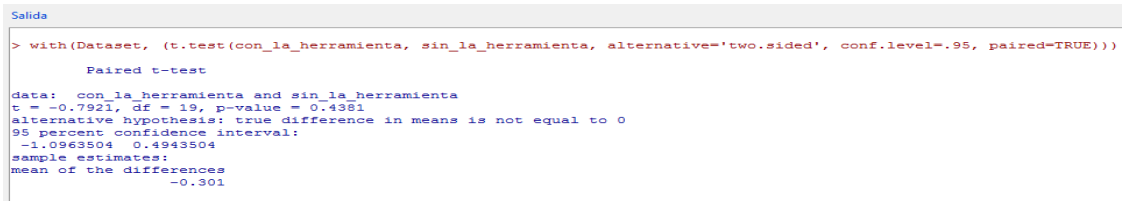

Fuente: elaboración propia

*Ilustración 50: Comando para el test T*




Fuente: elaboración propia

*Ilustración 51: Prueba de hipótesis - Velocidad de Navegación*




Fuente: elaboración propia

**Análisis**: Con la aceptación de un error igual a 0.05 equivalente al 95% de acierto, y con un p valor de 0.4386 mayor al nivel de significancia de 0.05, se concluye que la velocidad de navegación con el uso de la herramienta no se ve afectada negativamente, según el análisis del experimento, este comportamiento se justifica por la caché del api de isolación, dado que, al visitar 20 veces el mismo sitio, la primera vez lo trae completo de internet para el primer cliente (el de la medición más alta con la herramienta), mientras que para los restantes, al ya haberlo visitado, lo entrega desde su caché, lo que reduce así ampliamente el tiempo de respuesta.

## 3.2. Resultados estadísticos de la hipótesis de la investigación

**HIPOTESIS NULA Ho:** La implementación de una herramienta accesible de *Web Isolation*, **no** permitirá disminuir los costos del aseguramiento de los equipos clientes.

**HIPOTESIS ALTERNATIVA Ha:** La implementación de una herramienta accesible de *Web Isolation*, permitirá disminuir los costos del aseguramiento de los equipos clientes.

NIVEL DE SIGNIFICANCIA: $p < 0.05$.

### 3.2.1. Comparativa de costos.

En consideración a los modelos de negocio más populares, se establecen 2 orígenes de costos frecuentes, costo de licenciamiento y costo de operación, por cuanto la

primera puesta en marcha (implementación) incluirá este costo mientras que el mantenimiento será especificado por el precio de uso de la herramienta.

Se consideraron herramientas de la competencia de las cuales se logró obtener información de precios de medios confiables y oficiales.

*Tabla 12: Medición variable dependiente costos de implementación*

| Proveedor | Costo implementación | Referencias |
|---|---|---|
| Kasm | $ 0.00 | (Kasm, 2021) |
| Citrix Secure Browser | $ 0.00 | (citriz, 2021) |
| Cigloo | $ 0.00 | (Sourceforge, 2019) |
| ViewFinder | $ 0.00 | (dosyago, 2020) |
| Passage | $ 0.00 | (g2.com, 2021) |
| Webgap | $ 0.00 | (Webgap, 2019) |
| Solución Propuesta | $ 0.00 | - |
| **Promedio** | $ 0.00 | - |

Fuente: elaboración propia

Como se observa en la Tabla 12, no hay costo de implementación de las herramientas existentes en el mercado, así como el de la solución propuesta por lo que para el análisis de costos está variable es despreciable.

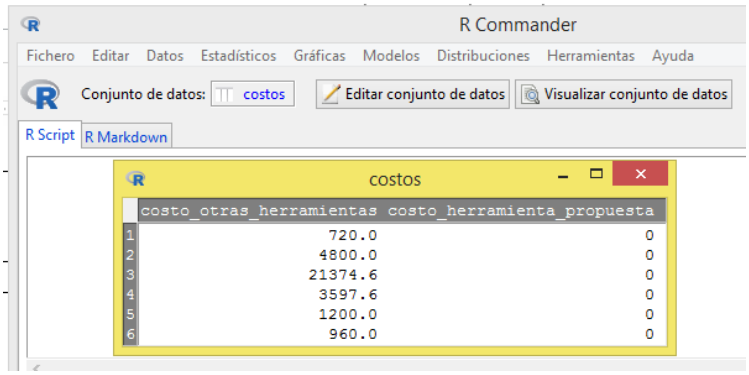
Para la estimación del costo de mantenimiento, se considera un total de 20 clientes al igual que en los experimentos anteriores y el lapso de tiempo establecido es de 12 mes que es el período regular contable, información que se detalla en la Tabla 13.

*Tabla 13: Medición variable dependiente costos de mantenimiento*

| Proveedor | Costo mensual por usuario | Cantidad de usuario | Total anual | Referencias |
|---|---|---|---|---|
| Kasm | $ 4.00 | 20 | $ 960.00 | (Kasm, 2021) |
| Citrix Secure Browser | $ 3.00 | 20 | $ 720.00 | (citriz, 2021) |
| Cigloo | $ 20.00 | 20 | $ 4800.00 | (Sourceforge, 2019) |
| ViewFinder | $ 89.06 | 20 | $ 21374.60 | (dosyago, 2020) |
| Passage | $ 14.99 | 20 | $ 3597.60 | (g2.com, 2021) |
| Webgap | $ 5.00 | 20 | $ 1200.00 | (Webgap, 2019) |
| Solución Propuesta | $ 0.00 | 20 | $ 0.00 | - |
| **Promedio** | $22.68 | 20 | $ 5441.93 | - |

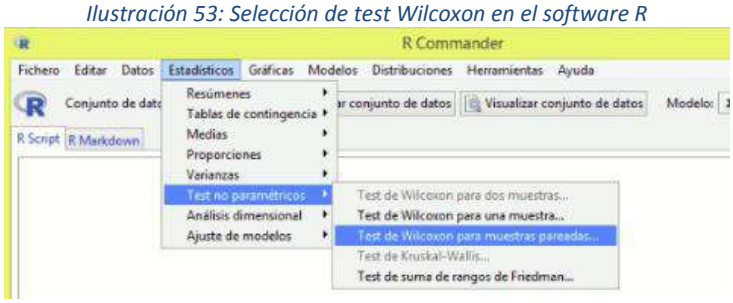
Fuente: elaboración propia

*Ilustración 52: Datos costos de implementación en el software estadístico R*



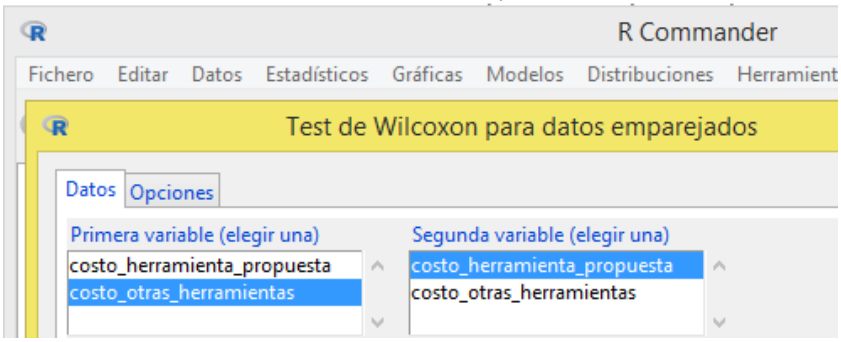

Fuente: elaboración propia

*Ilustración 53: Selección de test Wilcoxon en el software R*



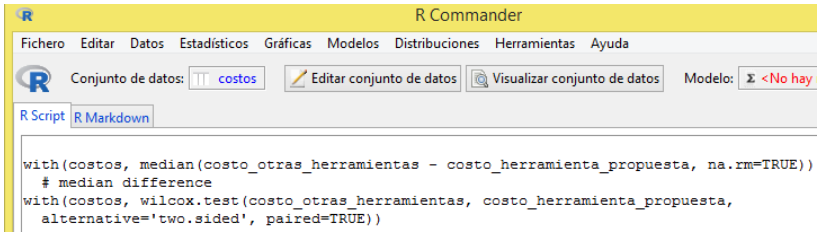

Fuente: elaboración propia

*Ilustración 54: Selección de Variables para el test Wilcoxon*




Fuente: elaboración propia

*Ilustración 55: Comando para el test Wilcoxon en R*




Fuente: elaboración propia

*Ilustración 56: Prueba de hipótesis - Costos de implementación*

```
        Wilcoxon signed rank exact test

data:  costos1$otras_herramientas and costos1$herramienta_propuesta
V = 15, p-value = 0.03125
alternative hypothesis: true location shift is greater than 0
```

Fuente: elaboración propia

**Análisis:** Con la aceptación de un error igual a 0.05 equivalente al 95% de acierto, con un p valor igual a 0.03125 menor al nivel de significancia del 0.05, se decide aceptar la hipótesis alternativa: “La implementación de una herramienta accesible de *Web Isolation*, permitirá disminuir los costos del aseguramiento de los equipos clientes”, lo que permite concluir que el costo de la herramienta propuesta permite un ahorro de entre 720 dólares hasta 21374.60 dólares anuales.

## CONCLUSIONES

1. La fundamentación teórica del uso de las herramientas de Web *Isolation* existentes, para el desarrollo de la investigación, permitió realizar un fork del proyecto a utilizar al disponer de la versión final de browsergap antes de que el proyecto fuera renombrado como ViewFinder y cambiar su licencia de distribución, por cuanto se pudo modificarlo sin infringir ninguna normativa.
2. Con la definición de los requerimientos mínimos para la herramienta de Web *Isolation* en la disminución de los costos del aseguramiento de la navegación de los equipos clientes, se logró desarrollar una herramienta con $0,0 costo de instalación y mantenimiento, permitiendo con ello el acceso a soluciones de seguridad a empresas con limitados presupuestos.
3. El diseño de la solución Web *Isolation* como mecanismo de seguridad frente a las amenazas web en los equipos clientes, demostró en pruebas un aislamiento del 100% en relación a malware distribuido por sitios web con objetivos del sistema operativo, así mismo, una capacidad de recuperación inmediata hacia ataques dirigidos al navegador distribuidos por sitios web.

## RECOMENDACIONES

1. Para futuras investigaciones se recomienda probar el uso del protocolo WebRTC para la transmisión del sitio renderizado en lugar del *stream* de imágenes que utiliza la presente solución.
2. Se considera los inconvenientes presentados en las iteraciones 7 y 8, se recomienda probar alternativas como traefik u otras disponibles para tratar de complementar el soporte https.
3. Al observar el pre procesamiento de la navegación encontrado en el navegador chrome, se recomienda probar la herramienta con el navegador open source chromium con la finalidad de modificar el código fuente del mismo para evitar este proceso.

**BIBLIOGRAFÍA**

Avfirewalls. (2020). *Fortinet FortiIsolator 1000F | AVFirewalls.com*. https://www.avfirewalls.com/FortiIsolator-1000F.asp

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas., D. (2001). *Manifiesto por el Desarrollo Ágil de Software*. http://agilemanifesto.org/iso/es/manifesto.html

citriz. (2021). *Cloud-Based Virtual Browser for Secure Web Browsing - Citrix*. https://www.citrix.com/products/citrix-secure-browser/

Digital.ai. (2020). *14th Annual State of Agile Report*. Digital.Ai. https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494

dosyago. (2020). *Dosyago*. https://dosyago.com/

g2.com. (2021). *Passages Pricing 2021 | G2*. https://www.g2.com/products/passages/pricing

Kasm. (2021). *Kasm Server*. https://www.kasmweb.com/kasm_server.html

Kaspersky. (2019). *Amenazas web | Malware de navegador de Internet | Kaspersky*. https://latam.kaspersky.com/resource-center/threats/web

MacDonald, N. (2016). *It's Time to Isolate Your Users From the Internet Cesspool With Remote Browsing*. https://doi.org/G00315285

McAFee. (2020). *What is Browser Isolation? | Light Point Security*. https://lightpointsecurity.com/what-is-browser-isolation

Norton. (2020). *Jigsaw Ransomware quiere jugar un juego, pero no de una buena manera*. https://us.norton.com/internetsecurity-emerging-threats-jigsaw-ransomware-wants-to-play-a-game-but-not-in-a-good-way.html

Osterman, M. (2018). *Why You Should Seriously Consider Web Isolation Technology*. https://www.mcafee.com/enterprise/en-us/assets/white-papers/wp-osterman-

web-isolation-technology.pdf

Paúl, L. (2016, November 21). *Caso de Target y Riesgo de Cybersecurity - Diario Financiero*. Diario Financiero. https://www.df.cl/noticias/opinion/columnistas/caso-de-target-y-riesgo-de-cybersecurity/2016-11-20/195850.html

Perekalin, A. (2017, October 24). *Bad Rabbit: una nueva epidemia de ransomware va en aumento | Blog oficial de Kaspersky*. Kaspersky.Com. https://www.kaspersky.com/blog/bad-rabbit-ransomware/19887/

Red Hat. (2021). *¿Qué es el open source?* https://www.redhat.com/es/topics/open-source/what-is-open-source

Ron, R., & Sacoto, V. (2017). Las PYMES ecuatorianas: su impacto en el empleo como contribución del PIB PYMES al PIB total Ecuadorian SMEs: their impact on employment as a contribution of SME GDP to total GDP. In *Pág* (Vol. 38). https://www.revistaespacios.com/a17v38n53/a17v38n53p15.pdf

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*.

Slepogin, N. (2017, May 25). *Dridex: una historia de evolución | Securelist*. Kaspersky. https://securelist.com/dridex-a-history-of-evolution/78531/

Sourceforge. (2019). *Cigloo Browser Isolation Management Platform Reviews and Pricing 2020*. https://sourceforge.net/software/product/Cigloo-Browser-Isolation-Management-Platform/

Valinsk, J. (2018, August 1). *Three people arrested for massive Chipotle, Arby's, Chili's hacks*. CNN. https://money.cnn.com/2018/08/01/technology/fin7-hackers-arrested/index.html

Webgap. (2019). *Remote Browser Pricing*. https://webgap.io/remote-browser-isolation-pricing.html

zscaler. (2020). *¿Qué es el aislamiento remoto del navegador? | Definición y conceptos*. https://www.zscaler.com/resources/security-terms-glossary/what-is-remote-browser-isolation

# ANEXOS

## Anexo 1: Historias de Usuario

*Tabla 14: Historia de Usuario 1*

| HISTORIA DE USUARIO | |
|---|---|
| **N°:** 1 | **Sprint:** 1 |
| **Nombre de la Historia:** Definición de la arquitectura de la solución *web isolation.* | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 07/09/2020 | **Fecha Fin:** 10/09/2020 |
| **Descripción:**<br>**Contexto** Dado que, se tienen requisitos específicos en cuanto a las funcionalidades requeridas y al rendimiento esperado, es necesario antes de iniciar definir una arquitectura que soporte todas las funcionalidades requeridas<br>**Como Desarrollador** necesito definir una arquitectura candidata **para** poder iniciar un desarrollo ordenado y viable.<br>**Criterios de aceptación**<br>1. El diagrama se encontraría en una herramienta de diagramada visual en línea.<br>2. Será compuesta exclusivamente por herramientas de software libre.<br>3. Permitir la escalabilidad. | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 1:

1. Análisis de herramientas de diagramado visual.
2. Análisis de herramientas DNS disponibles.
3. Estructuración de puertos.
4. Definición de herramientas a desarrollar.

*Tabla 15: Historia de Usuario 2*

| HISTORIA DE USUARIO | |
|---|---|
| **N°:** 2 | **Sprint:** 1 |
| **Nombre de la Historia:** Preparación del ambiente de desarrollo. | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 11/09/2020 | **Fecha Fin:** 16/09/2020 |
| **Descripción:**<br>**Contexto** Dado que, se requieren diferentes tecnologías para la consecución del proyecto, es necesario disponer de un ambiente que integre las mismas de manera local para su uso<br>**Como Desarrollador** necesito con todas las herramientas necesarias **para** poder realizar el desarrollo de la solución.<br>**Criterios de aceptación**<br>1. Se utilizarán versiones open source o en su defecto comunitarias de los entornos<br>2. No se utilizará ninguna herramienta que implique un costo por su uso<br>3. Se usa versiones instalables tanto como web mientras cumplan las condiciones anteriores. | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 2:

1. Selección de un IDE para el *backend.*
2. Selección de un IDE para el *front end.*
3. Selección de un repositorio.
4. Inicialización de los repositorios.
5. Ambiente docker local.

*Tabla 16: Historia de Usuario 3*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 3 | **Sprint:** 2 |
| **Nombre de la Historia:** Despliegue de la infraestructura para pruebas | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 17/09/2020 | **Fecha Fin:** 21/09/2020 |
| **Descripción:**<br>**Contexto** Dado que la solución se compone de varios elementos de software, se requiere un entorno integrado para la ejecución y prueba conjunta de los mismos.<br>**Como Desarrollador** necesito una infraestructura de pruebas **para** poder realizar experimentos controlados de la solución.<br>**Criterios de aceptación**<br>1. Se usaría recursos locales o remotos.<br>2. Se requiere que se ejecute sobre Linux | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 3:

1. Instalación de docker-compose.
2. Pruebas de redes y volúmenes.

*Tabla 17: Historia de Usuario 4*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 4 | **Sprint:** 2 |
| **Nombre de la Historia:** Implementación de servidor DnsMasq | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 22/09/2020 | **Fecha Fin:** 02/10/2020 |
| **Descripción:**<br>**Contexto** Dado que se requiere aislar determinados sitios web, se requiere una manera de redireccionar las peticiones de los mismos a nuestro servidor de isolación.<br>**Como Usuario** necesito que se redirijan mis peticiones de sitios aislados al servidor de isolación **para** prevenir incidentes de seguridad.<br>**Criterios de aceptación**<br>1. El servidor DNS será accesible desde los hosts físicos de la red<br>2. La resolución externa estaría configurada hacia los DNS de google y cloudflare en este orden de prioridad | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 4:

1. Despliegue del servicio a través de docker-compose.
2. Configuración del servidor.
3. Pruebas de resolución.

*Tabla 18: Historia de Usuario 5*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 5 | **Sprint: 3** |
| **Nombre de la Historia:** Implementación de un servidor de aislamiento | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 03/10/2020 | **Fecha Fin:** 18/10/2020 |
| **Descripción:**<br>**Contexto** Dado que, se requiere aislar la navegación web de ciertos sitios, se requiere un servicio capaz de procesar esos sitios y entregar el render a los clientes finales.<br>**Como Usuario** necesito poder navegar en los sitios aislados<br>**Criterios de aceptación**<br>1. El sitio aislad abre en la misma pestaña que lo solicitó<br>2. No sería necesaria ninguna interacción adicional para usar el sitio aislado<br>3. La herramienta sera Open Source y tener su código completo disponible para poder modificarlo | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 5:

1. Análisis y selección de una herramienta.
2. Quitar la funcionalidad de agregar pestañas.
3. Cambiar funcionalidad de verificación de SessionId.
4. Modificar la navegación hacia adelante y hacia atrás dentro del sitio aislado.
5. Quitar la funcionalidad para modificar la URL.
6. Re ajustar el canvas para optimizar el espacio disponible en pantalla.

*Tabla 19: Historia de Usuario 6*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 6 | **Sprint: 4** |
| **Nombre de la Historia:** Plataforma web para la gestión de sitios a aislar | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 19/10/2020 | **Fecha Fin:** 26/10/2020 |
| **Descripción:**<br>**Contexto** Dado que, el servidor DNS presenta configuraciones complejas y por terminal, se requiere facilitar su configuración a través de una interfaz web<br>**Como Administrador de la herramienta** necesito poder gestionar los sitios aislados **para** el mantenimiento de la herramienta.<br>**Criterios de aceptación**<br>1. Se valida las direcciones ip asignadas como destino | |

| |
|---|
| 2. La dirección de destino será modificada<br>3. Se realiza el Agregar, Modificar o Eliminar todos sitios del listado |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 6:

1. Rest Api para el control del *back end.*
2. Crear un repository para el acceso a datos basado en un archivo de texto.
3. Listar todos los sitios aislados.
4. Crear un nuevo sitio.
5. Modificar un sitio.
6. Eliminar un sitio.
7. Web *front end*.

*Tabla 20: Historia de Usuario 8*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 8 | **Sprint: 5** |
| **Nombre de la Historia:** Sesiones independientes en el servidor de isolación | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 04/11/2020 | **Fecha Fin:** 08/11/2020 |
| **Descripción:**<br>**Contexto** Dado que el servidor centralizado de aislamiento atenderá las peticiones de todos los equipos miembros del entorno, se requiere que cada uno reciba únicamente las respuestas a sus peticiones.<br>**Como Usuario** necesito diferenciar mis peticiones y mi tráfico **para** mantener mi privacidad.<br>**Criterios de aceptación**<br>1. Por cada usuario que acceda se generaría y almacenaría un token de sesión independiente<br>2. Los tokens durarán 10 minutos como máximo para evitar saturación | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 8:

1. Modificación para autogenerar tokens de sesión.
2. Agregar caducidad del token.
3. Proceso de limpieza de tokens.

*Tabla 21: Historia de Usuario 9*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 9 | **Sprint: 5** |
| **Nombre de la Historia:** Bloqueo de creación nuevas pestañas dentro del navegador aislado | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 09/11/2020 | **Fecha Fin:** 19/11/2020 |
| **Descripción:** | |

| |
|---|
| **Contexto** Dado que la navegación aislada se definió solo para ciertos sitios, es necesario bloquear esta característica del navegador interno<br>**Como Administrador de la herramienta** necesito bloquear la creación de nuevas pestañas por los usuarios dentro del navegador aislado **para** evitar que aíslen sitios no autorizados.<br>**Criterios de aceptación**<br>1. No existiría la opción visual de agregar pestañas nuevas dentro del navegador aislado<br>2. Tampoco permitir agregar una pestaña por atajos de teclado |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 9:

1. Modificar la hoja de estilos.
2. Retirar la función javascript.
3. Modificar el interceptor de shotcuts.

*Tabla 22: Historia de Usuario 10*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 10 | **Sprint: 6** |
| **Nombre de la Historia:** Gestión de contenido multimedia | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 20/11/2020 | **Fecha Fin:** 24/11/2020 |
| **Descripción:**<br>**Contexto** Dado que, requiero acceder a sitios aislados de descarga de documentos, requiero poder visualizar archivos PDF o de imagen dentro del navegador aislado<br>**Como Usuario** necesito poder acceder a archivos multimedia desde el navegador aislado **para** observar su contenido.<br>**Criterios de aceptación**<br>1. Los documentos pdf se abrirán en el mismo navegador y NO en una pestaña nueva<br>2. Las imágenes serán renderizadas inline y NO se permitirá su copia o descarga | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 10:

1. Modificar la función javascript que valida el *token* de suscripción para esta funcionalidad.
2. Completar la función de renderizado.
3. Habilitar en el menú contextual la opción de “Almacenar como captura de pantalla”.

*Tabla 23: Historia de Usuario 11*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 11 | **Sprint: 6** |
| **Nombre de la Historia:** Estadísticas de comportamiento de la navegación aislado | |

| | |
|---|---|
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 25/11/2020 | **Fecha Fin:** 05/12/2020 |
| **Descripción:**<br>**Contexto** Dado que, se requieren realizar mediciones del comportamiento se requiere obtener información sobre la navegación.<br>**Como Desarrollador** necesito métricas de funcionamiento de la navegación **para** comparar y optimizar el rendimiento.<br>**Criterios de aceptación**<br>1. Se medirá la cantidad de tráfico generado y el tiempo de respuesta<br>2. Será visible en la interfaz mientas se navega | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 11:

1. Agregar un interceptor para poder medir los tiempos.
2. Implementar un provider que mantenga la información disponible a través de toda la herramienta.
3. Visualizar en el front end la información del provider en tiempo real.

*Tabla 24: Historia de Usuario 12*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 12 | **Sprint: 7** |
| **Nombre de la Historia:** Implementación de Reverse Proxy para redirección del puerto 80 al del API de *Web Isolation* | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 06/12/2020 | **Fecha Fin:** 21/12/2020 |
| **Descripción:**<br>**Contexto** Dado que, la redirección DNS se realiza directamente a una ip, la misma se resuelve en el puerto 80, pero la herramienta de isolación se encuentra en un puerto diferente, además, que se requiere manipular la petición para enviar la url solicitada como parámetro al API, se requiere un reverse proxy que tenga esa responsabilidad<br>**Como Desarrollador** necesito un reverse proxy **para** poder manipular y redireccionar las peticiones hacia el servidor de isolación.<br>**Criterios de aceptación**<br>1. La url solicitada sería reenviada en el parámetro “target” al api de isolación<br>2. Se integraría al docker-compose de la solución | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 12:

1. Agregar el servicio de nginx al docker-compose.
2. Configurar redirección de la urldsd.

*Tabla 25: Historia de Usuario 13*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 13 | **Sprint: 8** |
| **Nombre de la Historia:** Reemplazar reverse proxy por un software propio encargado de la redirección | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 01/03/2021 | **Fecha Fin:** 0503/20 |
| **Descripción:**<br>**Contexto** Dado que, los sitios con certificados estrictos no logran ser re direccionados pro el reverse proxy, se requiere desarrollar un api de redirección encargado de este proceso.<br>**Como usuario** necesito poder isolar todos los sitios https.<br>**Criterios de aceptación**<br>1. El api funcionará en docker<br>2. El api funcionará en el puerto 80 para poder integrarse en el entorno del mismo compose | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 13:

4. Crear un servicio web capaz de recibir peticiones en el puerto 80.
5. Re direccionar peticiones hacia el api de isolación.
6. Integrar en el entorno del docker-compose.

*Tabla 26: Historia de Usuario 14*

| **HISTORIA DE USUARIO** | |
|---|---|
| **N°:** 14 | **Sprint: 9** |
| **Nombre de la Historia:** Diseño de una solución para aislar sitios https | |
| **Estado:** Terminado | **Responsable:** Ericka Guanoluisa |
| **Fecha de Inicio:** 08/03/2021 | **Fecha Fin:** 12/03/2021 |
| **Descripción:**<br>**Contexto** Dado que, ninguna de las opciones de redirección intentadas funciona correctamente, se requiere una alternativa para poder isolar sitios https.<br>**Como usuario necesito una alternativa para poder isolar sitios https**.<br>**Criterios de aceptación**<br>3. Deberá funcionar para cualquier sitio https<br>4. Deberá estar integrado al entorno compose de la solución | |

Fuente: elaboración propia

Lista de tareas relacionadas historia de usuario 14:

3. Crear una interfaz para ingresar la url a isolar
4. Construir la ruta necesaria para que la dirección funcione con el api
5. Integrar la solución al entorno del docker-compose

## Anexo 2: Datos de pruebas

Velocidad de navegación sin la herramienta 20 equipos

*Ilustración 57: Velocidad de navegación sin la herramienta equipo 1*


Fuente: elaboración propia

*Ilustración 58: Velocidad de navegación sin la herramienta equipo 2*


Fuente: elaboración propia

*Ilustración 59: Velocidad de navegación sin la herramienta equipo 3*


Fuente: elaboración propia

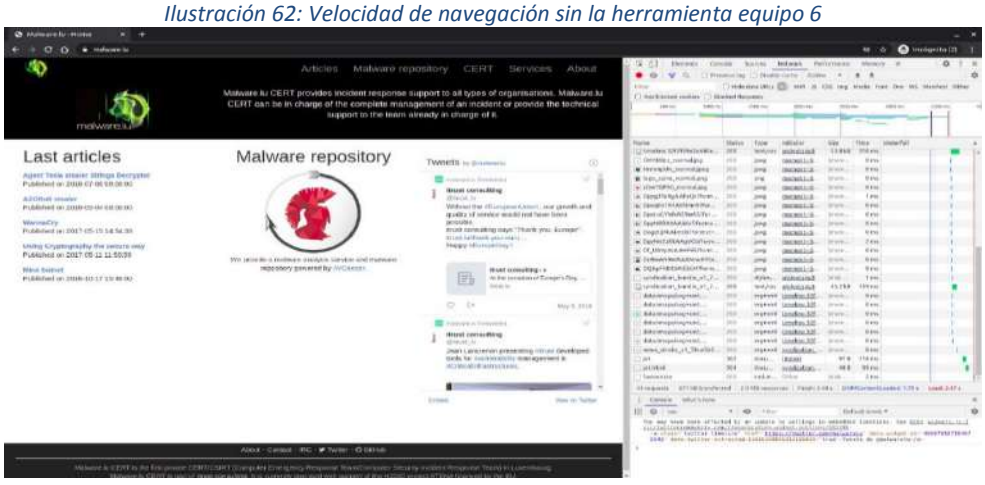
*Ilustración 60: Velocidad de navegación sin la herramienta equipo 4*


Fuente: elaboración propia

*Ilustración 61: Velocidad de navegación sin la herramienta equipo 5*


Fuente: elaboración propia

*Ilustración 62: Velocidad de navegación sin la herramienta equipo 6*

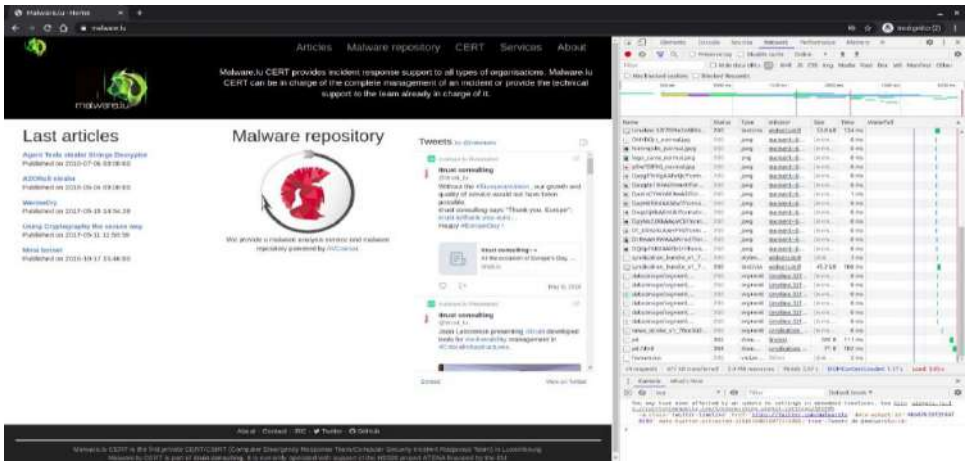



Fuente: elaboración propia

*Ilustración 63: Velocidad de navegación sin la herramienta equipo 7*


Fuente: elaboración propia

*Ilustración 64: Velocidad de navegación sin la herramienta equipo 8*




Fuente: elaboración propia

*Ilustración 65: Velocidad de navegación sin la herramienta equipo 9*


Fuente: elaboración propia

*Ilustración 66: Velocidad de navegación sin la herramienta equipo 10*


Fuente: elaboración propia

*Ilustración 67: Velocidad de navegación sin la herramienta equipo 11*


Fuente: elaboración propia

*Ilustración 68: Velocidad de navegación sin la herramienta equipo 12*


Fuente: elaboración propia

*Ilustración 69: Velocidad de navegación sin la herramienta equipo 13*


Fuente: elaboración propia

*Ilustración 70: Velocidad de navegación sin la herramienta equipo 14*


Fuente: elaboración propia

*Ilustración 71: Velocidad de navegación sin la herramienta equipo 15*


Fuente: elaboración propia

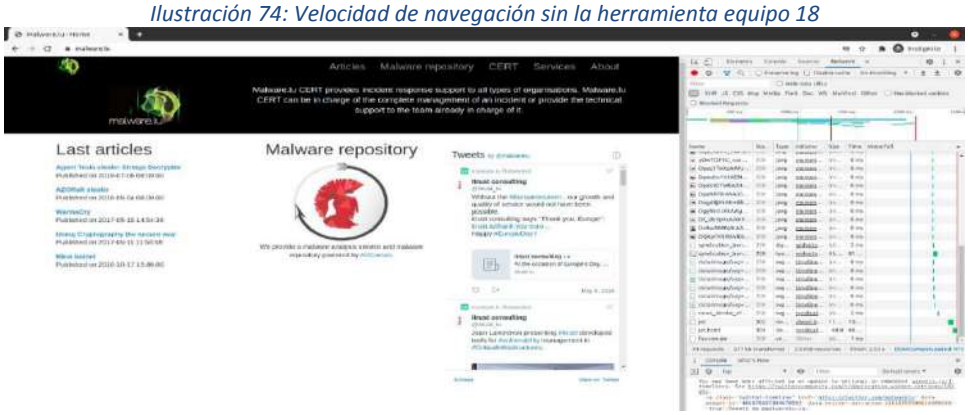
*Ilustración 72: Velocidad de navegación sin la herramienta equipo 16*


Fuente: elaboración propia

*Ilustración 73: Velocidad de navegación sin la herramienta equipo 17*


Fuente: elaboración propia

*Ilustración 74: Velocidad de navegación sin la herramienta equipo 18*




Fuente: elaboración propia

*Ilustración 75: Velocidad de navegación sin la herramienta equipo 19*

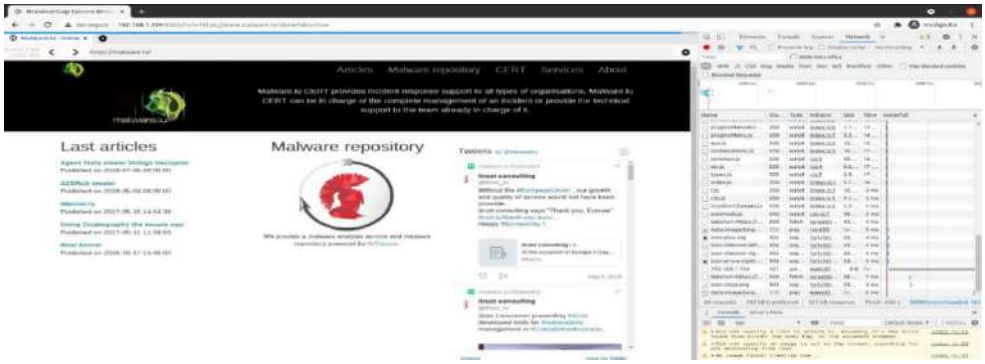

Fuente: elaboración propia

*Ilustración 76: Velocidad de navegación sin la herramienta equipo 20*


Fuente: elaboración propia

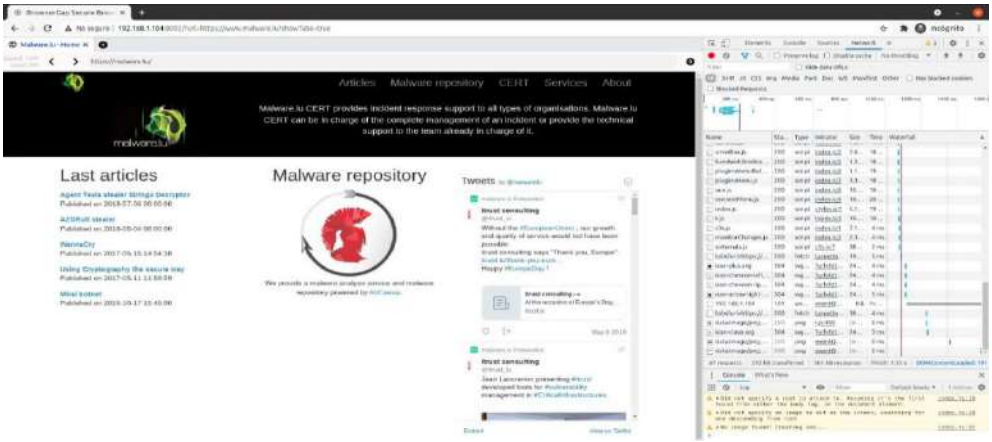
Velocidad de navegación con la herramienta 20 equipos

*Ilustración 77: Velocidad de navegación con la herramienta equipo 1*



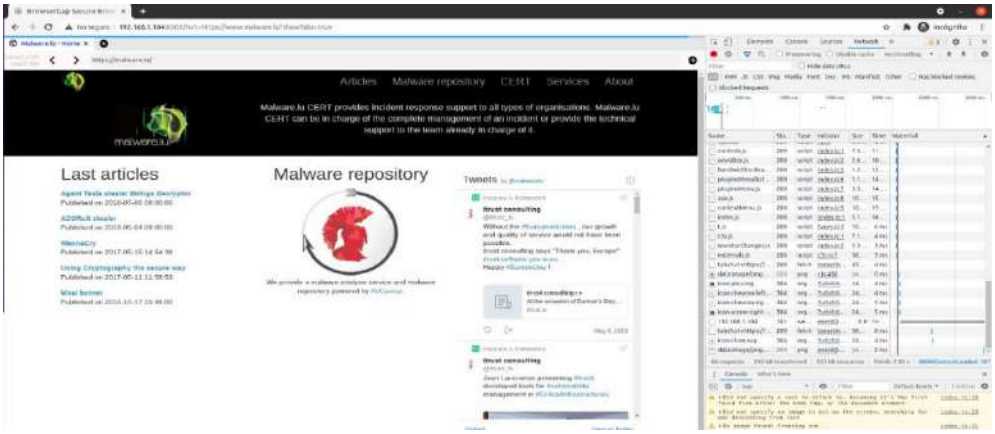

Fuente: elaboración propia

*Ilustración 78: Velocidad de navegación con la herramienta equipo 2*




Fuente: elaboración propia

*Ilustración 79: Velocidad de navegación con la herramienta equipo 3*

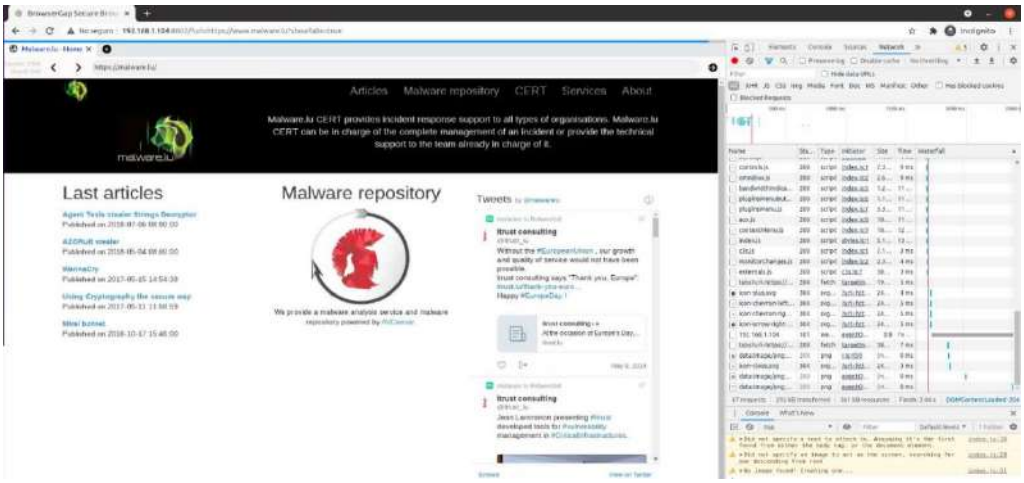



Fuente: elaboración propia

*Ilustración 80: Velocidad de navegación con la herramienta equipo 4*


Fuente: elaboración propia

*Ilustración 81: Velocidad de navegación con la herramienta equipo 5*




Fuente: elaboración propia

*Ilustración 82: Velocidad de navegación con la herramienta equipo 6*


Fuente: elaboración propia

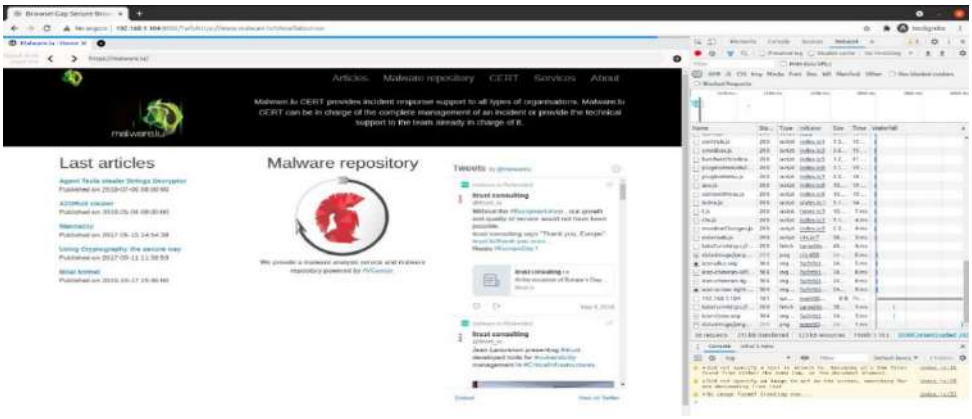
*Ilustración 83: Velocidad de navegación con la herramienta equipo 7*


Fuente: elaboración propia

*Ilustración 84: Velocidad de navegación con la herramienta equipo 8*

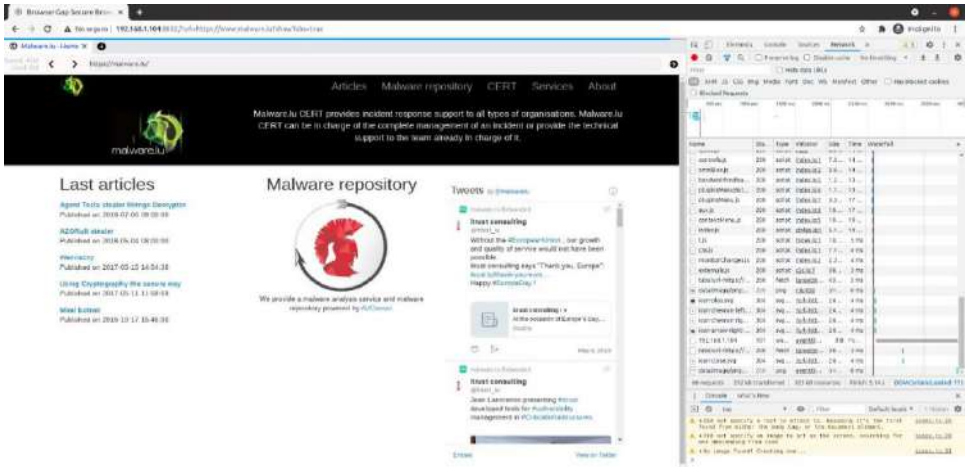

Fuente: elaboración propia

*Ilustración 85: Velocidad de navegación con la herramienta equipo 9*




Fuente: elaboración propia

*Ilustración 86: Velocidad de navegación con la herramienta equipo 10*

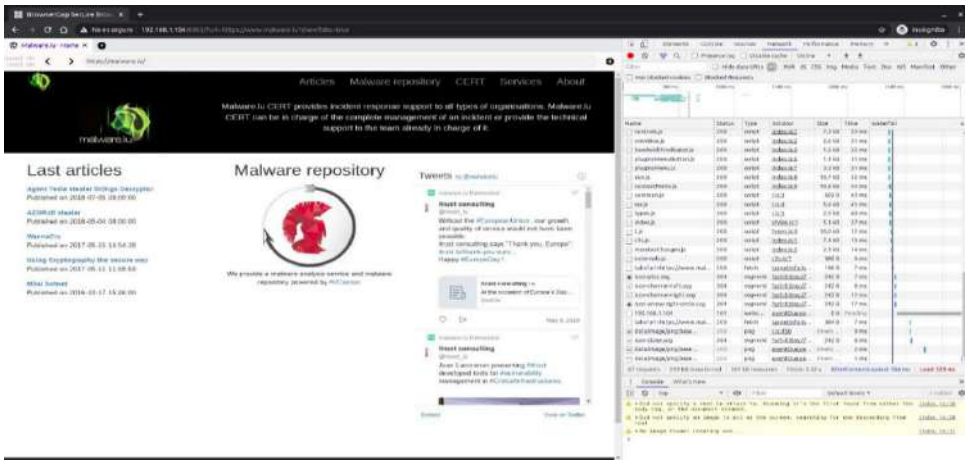



Fuente: elaboración propia

*Ilustración 87: Velocidad de navegación con la herramienta equipo 11*

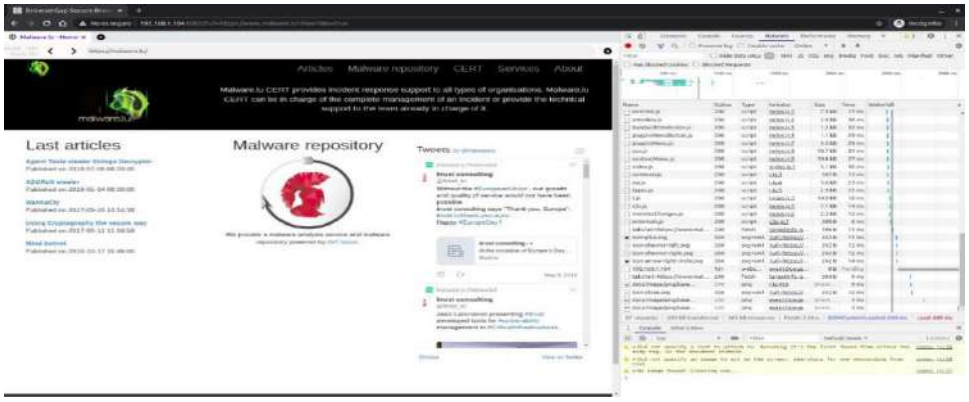

Fuente: elaboración propia

*Ilustración 88: Velocidad de navegación con la herramienta equipo 12*



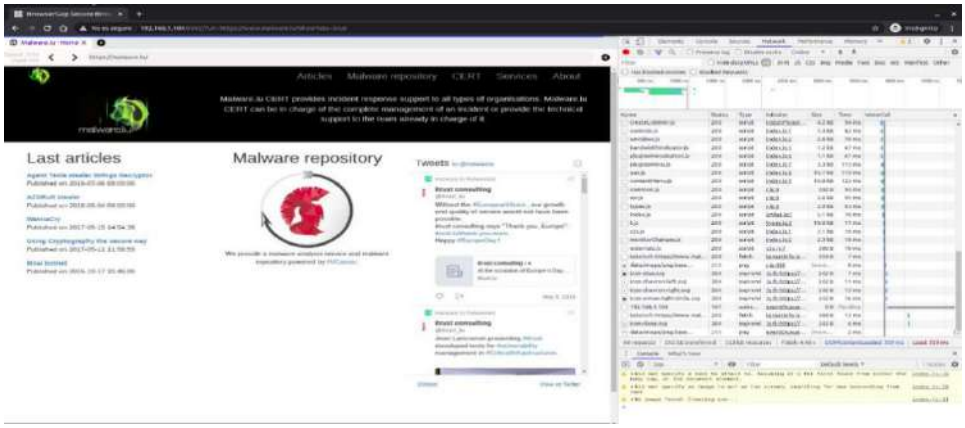

Fuente: elaboración propia

*Ilustración 89: Velocidad de navegación con la herramienta equipo 13*



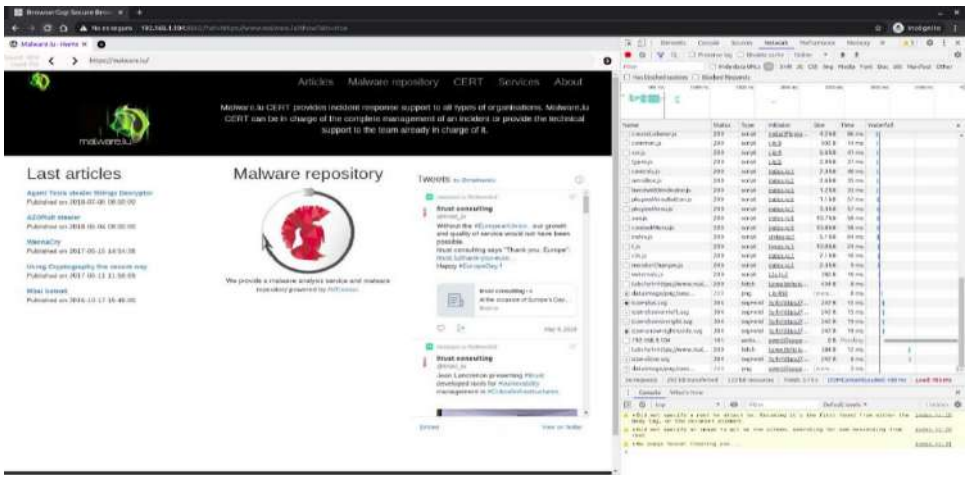

Fuente: elaboración propia

*Ilustración 90: Velocidad de navegación con la herramienta equipo 14*



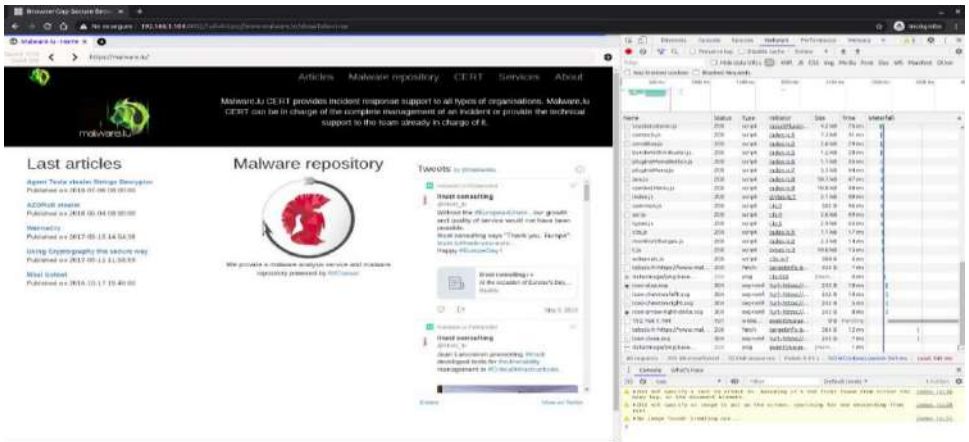

Fuente: elaboración propia

*Ilustración 91: Velocidad de navegación con la herramienta equipo 15*



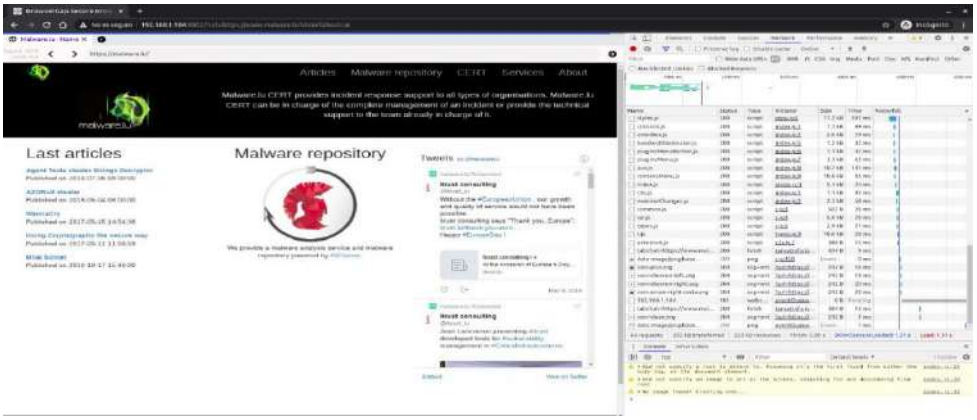

Fuente: elaboración propia

*Ilustración 92: Velocidad de navegación con la herramienta equipo 16*



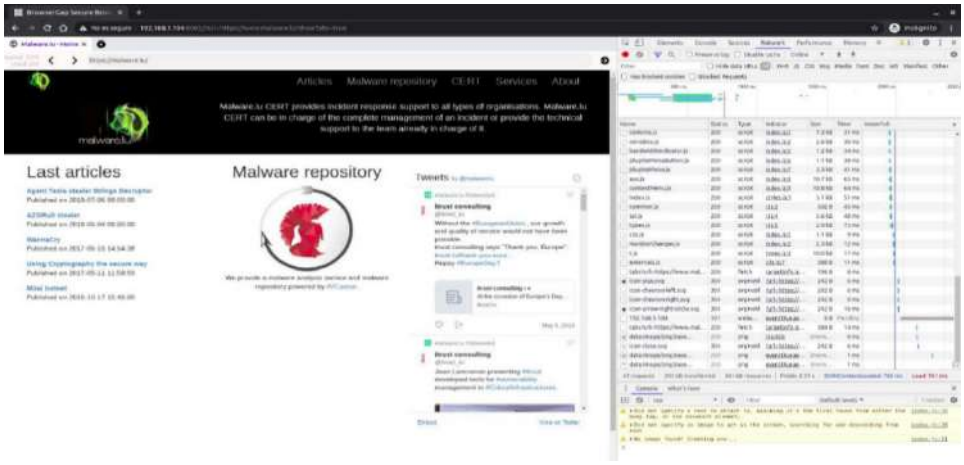

Fuente: elaboración propia

*Ilustración 93: Velocidad de navegación con la herramienta equipo 17*



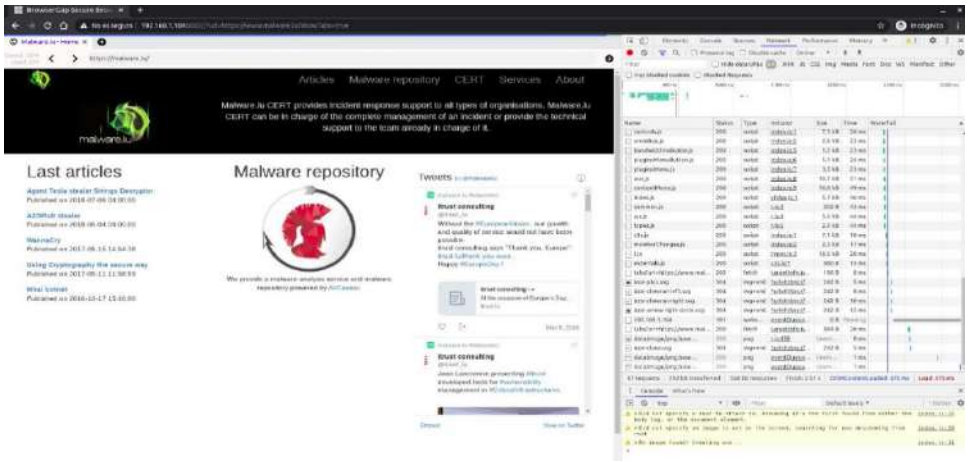

Fuente: elaboración propia

*Ilustración 94: Velocidad de navegación con la herramienta equipo 18*



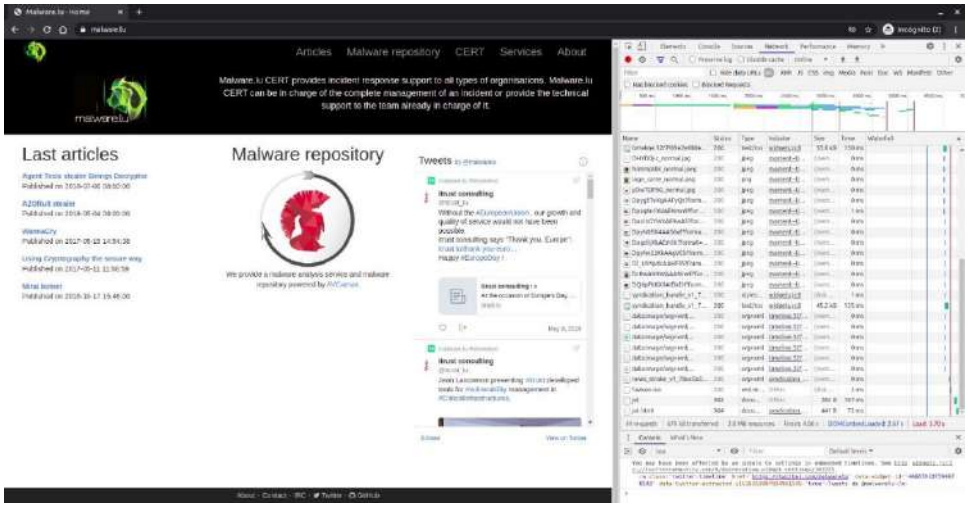

Fuente: elaboración propia

*Ilustración 95: Velocidad de navegación con la herramienta equipo 19*




Fuente: elaboración propia

*Ilustración 96: Velocidad de navegación con la herramienta equipo 20*




Fuente: elaboración propia

## Anexo 3: Guía de Implementación y uso

### Servidor

Se requiere instalación previa de docker y docker-compose

Las instrucciones detalladas para la instalación de docker en linux se encuentran en la página oficial y se recomienda el uso del *convenience script* https://docs.docker.com/engine/install/ubuntu/#install-using-the-convenience-script

De la misma manera para docker-compose https://docs.docker.com/compose/install/

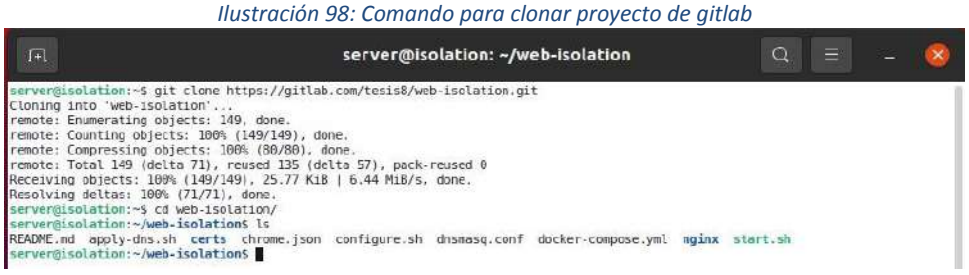
*Ilustración 97: Comando revisión de versión de docker*


Fuente: elaboración propia

Con los requisitos listos, se procede a clonar el repositorio mediante el comando: git clone https://gitlab.com/web-isolation.git

Con el proyecto clonado, ingresar a la carpeta web-isolation y verificar la existencia del archivo start.sh

*Ilustración 98: Comando para clonar proyecto de gitlab*




Fuente: elaboración propia

A continuación, se requiere configurar el ambiente en que se ejecutará el servicio, por cuanto se requiere mostrar los archivos ocultos para poder acceder al archivo .env que contiene las configuraciones necesarias.

Con cualquier editor de texto se colocaría tanto la IP como el puerto en el que se espera funcione el servidor de back end.

*Ilustración 99: Archivo de configuración .env*


Fuente: elaboración propia

Con el archivo preparado, se procede a configurar el servidor con el archivo configure.sh, mismo que serán ejecutados con permisos de super usuario, dado que, realizará las configuraciones mostradas en la siguiente imagen.

*Ilustración 100: Configuración archivo configure.sh*

```
server@isolation:~/web-isolation$ cat ./configure.sh
sudo su -c "echo 'kernel.unprivileged_userns_clone=1' > /etc/sysctl.d/00-local-userns.conf"
sudo su -c "echo 'net.ipv4.ip_forward=1' > /etc/sysctl.d/01-network-ipv4.conf"
sudo sysctl -p
sleep 1
echo 'DNS=1.1.1.1' | sudo tee -a /etc/systemd/resolved.conf
echo 'DNSStubListener=no' | sudo tee -a /etc/systemd/resolved.conf
sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
sudo systemctl restart systemd-networkd.service
sed -i 's/IP_ADDR_HERE/'$(head -1 .env | cut -b 12-30)'/g' ./nginx/conf.d/redirect.conf
server@isolation:~/web-isolation$ sudo sh ./configure.sh
[sudo] password for server:
DNS=1.1.1.1
DNSStubListener=no
server@isolation:~/web-isolation$
```

Fuente: elaboración propia

Con el equipo configurado, se inician todos los servicios con el script start.sh, el mismo construirá todo el ambiente necesario y los 5 servicios necesarios se ejecutarían correctamente.

*Ilustración 101: Ejecución del script start.sh*

```
server@isolation: ~/web-isolation
d7c30a071210: Pull complete
298db1a0cc00: Pull complete
75a27d47ac9e: Pull complete
Digest: sha256:d0563470068934e5c90867d6a9693d548759322179ea674abe0c3c808dcd283b
Status: Downloaded newer image for registry.gitlab.com/tesis8/web-isolation-manager-front-end:latest
Pulling reverse (nginx:)...
latest: Pulling from library/nginx
6f28985ad184: Pull complete
29f7ebf60efd: Pull complete
879a7c160ac6: Pull complete
de58cd48a671: Pull complete
be704f37b5f4: Pull complete
150aac73782c: Pull complete
Digest: sha256:d2925188effb4ddca9f14f162d6fba9b5fab232028aa07ae5c1dab764dca8f9f
Status: Downloaded newer image for nginx:latest
Creating reverse                                ... done
Creating web-isolation_dnsmasq_1                ... done
Creating web-isolation_web-isolation-manager_1 ... done
Creating web-isolation_nginx_1                  ... done
server@isolation:~/web-isolation$
```

Fuente: elaboración propia

Se verifica que los 5 servicios se encuentren en ejecución mediante los procesos de docker con el comando: docker ps

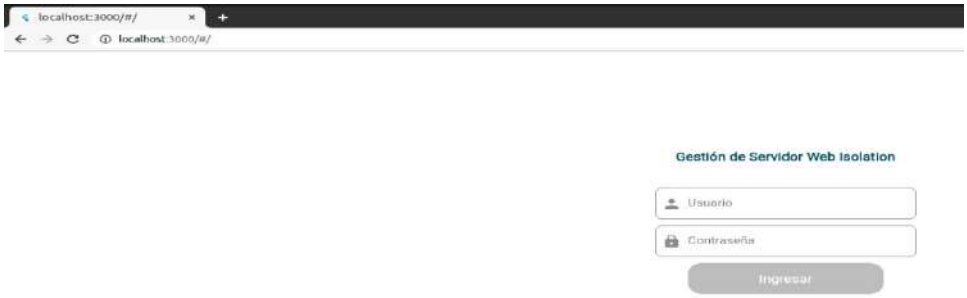
*Ilustración 102: Comando visualización de procesos docker*

```
server@isolation: ~
server@isolation:~$ docker ps
CONTAINER ID  IMAGE                                                                COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
1416b279ac5f  registry.gitlab.com/tesis8/web-isolation-manager-front-end            "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:3000->80/tcp                                   web-isolation_nginx_1
078e6d122c07  registry.gitlab.com/tesis8/web-isolation-manager:latest               "java -jar /app.jar"     2 minutes ago   Up 2 minutes   0.0.0.0:8080->8080/tcp                                 web-isolation_web-isolation-manager_1
178cf0f2545f  jpillora/dnsmasq                                                     "webproc --config /e…"   2 minutes ago   Up 2 minutes   0.0.0.0:53->53/udp, 0.0.0.0:5380->8080/tcp             web-isolation_dnsmasq_1
5feb55c6cd2f  nginx                                                                "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp               reverse
052a75e24aab  registry.gitlab.com/erickaguanoluisa92/isolation-api                 "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes   5002/tcp, 0.0.0.0:8002->8002/tcp                       isolation-api
server@isolation:~$
```

Fuente: elaboración propia

El puerto 3000 del servidor tendría ya en funcionamiento el administrador web de la herramienta.

*Ilustración 103: Página de inicio administrador web*




Fuente: elaboración propia

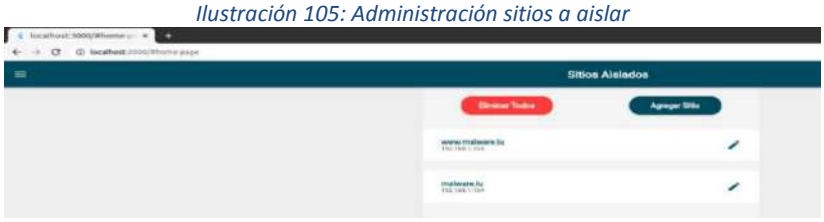
En el menú servidores se verificaría las direcciones de los isolation-api disponibles, por start.sh desplegará una instancia en el mismo servidor así que la dirección ipv4 sería la misma.

*Ilustración 104: Administración de servidor aislado*


Fuente: elaboración propia

En el menú sitios se configura todos aquellos sitios web que se requieran aislar con la herramienta y dirigirla al servidor correspondiente, por defecto estará disponible solo uno llamado principal.

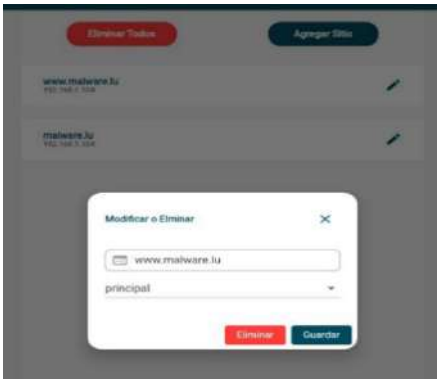
*Ilustración 105: Administración sitios a aislar*




Fuente: elaboración propia

Nota: La dirección web a aislar tiene la ruta exacta ya sea con o sin www, pero no contener el protocolo de comunicación ya sea http o https, dado que, el mismo será controlado por la herramienta.

*Ilustración 106: Administración sitios a aislar*




Fuente: elaboración propia

Una vez agregado los sitios web a aislar, se reinicia el servicio con el comando: sh ./apply-dns.sh

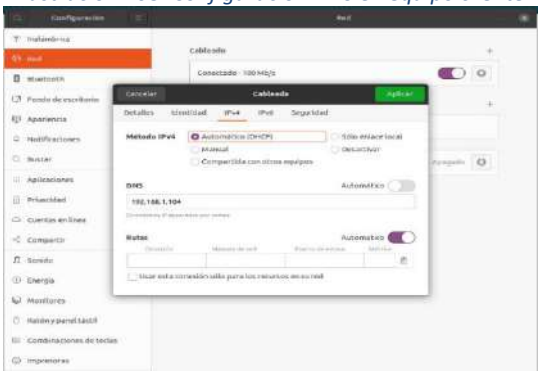
*Ilustración 107: Comando reinicio de servicio*


Fuente: elaboración propia

**<u>En los equipos Cliente</u>**

Configurar como DNS la IP del servidor *Web Isolation.*

*Ilustración 108: Configuración DNS en equipo cliente*



Fuente: elaboración propia

Los pasos para la instalación de la solución propuesta, también, se encuentra en el enlace: https://gitlab.com/tesis8/web-isolation/-/tree/http-only#installation