Desarrollo de un Prototipo de Framework para brindar seguridad en la confidencialidad de la información en el estándar HL7 CDA R2

Paulo Ernesto Diaz Ordoñez

Instituto Tecnológico Metropolitano
Facultad de Ingeniería
Medellín, Colombia
2020

Desarrollo de un Prototipo de Framework para brindar seguridad en la confidencialidad de la información en el estándar HL7 CDA R2

Paulo Ernesto Diaz Ordoñez

Tesis o trabajo de grado presentado como requisito parcial para optar al título de:
Magíster en Seguridad Informática

Director:
Mg Milton Javier Mateus

Línea de Investigación:
Manejo de incidentes de Seguridad y Análisis Forense
Instituto Tecnológico Metropolitano
Facultad de Ingeniería
Medellín, Colombia
2020

Dedicatoria

A mis Padres Paulo Emilio y Yoly Omary, dedico este esfuerzo y logro académico al ser cómplices de mis sueños y esperanzas, por enseñarme siempre con el mejor ejemplo a ser cada vez mejor ser humano dispuesto a servir y no claudicar jamás ante las adversidades de la vida.

A mis Hermanas Gloria Milena y Laurita Sofía, por soportarme y apoyarme en momentos difíciles de mi vida, por comprenderme integralmente; sus sabios consejos han permitido afrontar mi vida con humildad, sencillez y entrega al servicio de los demás.

"Todo hombre recibe dos educaciones: La que le dan y la que él se da; Esta última es la más Importante".

Edward Gibben.

Agradecimientos

A Dios y a mis padres gratitud eterna por ser principio, luz, ejemplo, abrigo y eterna fuerza que me anima a continuar forjando el futuro más promisorio para mi vida.

Serían incontables las experiencias y anécdotas vividas como estudiante de maestría en seguridad informática en el Instituto tecnológico Metropolitano de Medellín, debo reconocer que todas ellas estuvieron siempre permeadas de enormes sacrificios, de alegrías, de sufrimientos, de trasnochos, de lectura incansable, de vivencias únicas que fueron disfrutadas con el alma al tener la fortuna de encontrar maestros éticos, con convicción y compromiso, dispuestos incondicionalmente a compartir sus saberes y experiencias, por ello les expreso mi enorme gratitud, por ser incondicionales, altamente profesionales y auténticamente humanos, lo aprendido con ellos con seguridad acorta las distancias para alcanzar mis sueños, expectativas, intereses y esperanzas en mi desarrollo profesional y personal.

Agradecimientos especiales a mi Asesor de Tesis Mg Milton Javier Mateus por ser auténtico acompañante en mis estudios, guía, orientador y amigo, permitiéndome trazar rutas, encontrar caminos, reformular preguntas que hicieron posible alcanzar con éxito los objetivos y metas de la investigación.

A mis Compañeros los cuáles poco a poco se convirtieron en amigos, especialmente a Irving Solsol todo mi afecto y gratitud al ser parte integral de estas vivencias académicas y de compañerismo; compartir con él, conocimientos previos, proyectos, intereses comunes de investigación, desde el mismo momento de iniciar estudios, fortalecieron mi capacidad para trabajar en equipo y mi decisión de no claudicar ante las dificultades.

A mi Jefe de Programa Héctor Vargas mi aprecio y reconocimiento por abrirme puertas que me permitieron encontrar los profesionales claves para alivianar el camino y culminar con éxito mi Maestría.

Resumen

Se puede afirmar que dentro de las entidades de salud nace la necesidad de implementar un sistema de información electrónico para poder enviar datos e información, que en la mayoría de los casos es de carácter privado y confidencial; generalmente esta información es muy sensible para los pacientes porque contiene datos personales e historiales que registran enfermedades que padecen, datos que pueden ser utilizados negativamente si llegan a caer a manos de terceros como delincuentes informáticos, quienes tendrían toda posibilidad para robar su identidad e inclusive estar expuestos a altos riesgos que podrían incidir en la misma vida de los pacientes o en la perdida de confiabilidad de las entidades que utilizan dichos datos. Es por eso que la empresa HL7 pensando en los problemas de comunicación y envío de información de una entidad a otra en el sector salud crea un estándar llamado HL7 CDA(Clinical Document Architecture) R2 (Release 2) el cual permite realizar esta comunicación pensando en un carácter netamente de interoperabilidad, para que así se puedan realizar envíos de diferentes tipos de archivos y en diferentes plataformas, pero que puedan ser leídos por el sistema de otras entidades transmitiendo historiales médicos.

Es así como surge una pregunta: ¿Es seguro el Estándar HL7 CDA en su Confidencialidad?; partiendo de proyectos realizados hace algunos años se puede concluir que se ha trabajado muy poco en la seguridad de este estándar dejando ver algunos riesgos de seguridad que llevan a vulnerabilidades del estándar, de mucha importancia en la confidencialidad de los documentos que se transmiten por este medio, principalmente vulnerabilidades de XSS (Cross site scripting) y vulnerabilidades de obtención de información. Se entrega como resultado final del proyecto desarrollado un Prototipo de framework que permita brindar más seguridad en el estándar HL7 CDA R2, es por ello que se ha iniciado un estudio de varias herramientas de detección y prevención, de trabajo en conjunto bajo características fundamentales como, que sean OPEN SOURCE. Esta investigación se realizó bajo 3 fases metodológicas, primera fase Identificar problemas y Determinar Herramientas, donde se identificaron y caracterizaron los problemas de confidencialidad del estándar, se buscaron las mejores herramientas para detectar y prevenir los problemas que afectaron la confidencialidad del estándar. La segunda fase Desarrollo del prototipo de Framework, donde se recolectaron los historiales médicos electrónicos, se investigó acerca de un visor de documentos CDA web, se identificaron las medidas de seguridad para las vulnerabilidades encontradas y se simuló un envío de historiales médicos para su posterior análisis de vulnerabilidades. Para desarrollar el prototipo de framework se acoplo dos módulos (IDS, IPS) open source para trabajar en conjunto y de manera equilibrada, mostrando los ataques con su nivel de criticidad en score de CVE vulnerability. La tercera Fase es Evaluar el Prototipo de Framework Propuesto, donde se implementó el prototipo de framework en un servidor Ubuntu y se realizaron las pruebas pertinentes para corroborar que las vulnerabilidades fueron mitigadas, el desarrollo del prototipo se realizó con el fin de brindar confidencialidad de la información en los historiales

clínicos, los cuáles son transmitidos por este estándar.

Palabras clave: Análisis de vulnerabilidades,Ataque informático, CDA, Confidencialidad, Controles, cve, Framework, HL7, IDS, Interoperabilidad, IPS, owasp, PHVA, Sistemas de Información, Vulnerabilidad, wifi, xss.

Abstract

It can be affirmed that within the health entities the need arises to implement an electronic information system to be able to send data and information, which in most cases is private and confidential; This information is generally very sensitive for patients because it contains personal data and records that record illnesses that they suffer, data that can be used negatively if they fall into the hands of third parties such as computer criminals, who would have every possibility to steal their identity and even be exposed to high risks that could affect the life of the patients or the loss of reliability of the entities that use said data. That is why the company HL7 thinking about communication problems and sending information from one entity to another in the health sector creates a standard called HL7 CDA (Clinical Document Architecture) R2 (Release 2) which allows to carry out this communication thinking about a purely interoperability character, so that different types of files can be sent and on different platforms, but can be read by the system of other entities transmitting medical records.

This is how a question arises: Is the HL7 CDA Standard safe in its Confidentiality ?; Based on projects carried out a few years ago, it can be concluded that very little work has been done on the security of this standard, revealing some security risks that lead to vulnerabilities in the standard, of great importance in the confidentiality of the documents transmitted by this means. , mainly XSS (Cross site scripting) vulnerabilities and information gathering vulnerabilities. It is delivered as a final result of the project developed a prototype framework that allows to provide more security in the HL7 CDA R2 standard, that is why a study of various detection and prevention tools has begun, working together under fundamental characteristics such as , that are OPEN SOURCE. This research was carried out under 3 methodological phases, first phase Identify problems and Determine Tools, where the confidentiality problems of the standard were identified and characterized, the best tools were sought to detect and prevent the problems that affected the confidentiality of the standard. The second phase Development of the Framework prototype, where electronic medical records were collected, investigated about a web CDA document viewer, identified security measures for the vulnerabilities found, and simulated sending medical records for subsequent analysis. of vulnerabilities. To develop the framework prototype, two open source modules (IDS, IPS) were coupled to work together and in a balanced way, showing the attacks with their criticality level in the CVE vulnerability score. The third phase is to Evaluate the Proposed Framework Prototype, where the framework prototype was implemented on an Ubuntu server and the relevant tests were carried out to corroborate that the vulnerabilities were mitigated, the development of the prototype was carried out in order to provide confidentiality of the Information in the medical records, which are transmitted by this standard.

Keywords: CDA, Confidentiality, Controls, Computer attack, cve, Framework, HL7, IDS, Information Systems, Interoperability, owasp, IPS, PHVA, Vulnerability, Vulnerability analysis, wifi, xss.

Contenido

Agradecimientos vii

Resumen ix

1. Introducción 2

2. Marco Teórico y Estado del Arte 5

2.1. Marco teórico . . . 5

2.1.1. Sistema de Información en Salud. . . . 5

2.1.2. Interoperabilidad en Sistemas de Información en Salud. . . . 5

2.1.3. Estándar HL7 CDA R2 (Release 2) . . . 6

2.1.4. Análisis de Vulnerabilidades . . . 7

2.1.5. Modulo OSSEC . . . 7

2.1.6. Modulo ModSecurity . . . 8

2.1.7. Mirth Connect . . . 8

2.1.8. WHIRESHARK . . . 9

2.1.9. Man in the Middle (MitM) . . . 9

2.1.10. cross site scripting (xss) . . . 9

2.1.11. Open Web Application Security Project (OWASP) . . . 9

2.1.12. ¿Qué es un IDS y un IPS? . . . 10

2.1.13. ¿Qué es CVE? . . . 10

2.1.14. Framework . . . 10

2.2. Estado del Arte . . . 10

3. Metodología 13

3.1. Fase 1, Identificar problemas y Determinar Herramientas. . . . 14

3.2. Fase 2, Desarrollo del prototipo de Framework. . . . 19

3.3. Evaluación del Prototipo de Framework Propuesto. . . . 23

4. RESULTADOS 25

4.1. PRIMERA FASE, Identificar Problemas y Determinar Herramientas. . . . 25

4.1.1. Determinar Herramientas idóneas de detección y prevención de vulnerabilidades. . . . 35

4.2. SEGUNDA FASE, Desarrollo del Prototipo de Framework 36
4.2.1. Evaluar el Prototipo de Framework Propuesto. 47

5. Conclusiones y recomendaciones **53**
5.1. Conclusiones 53
5.2. Recomendaciones 54

A. Anexo: Instalación MIRTH CONNECT **56**

B. Anexo: Configuración y Creación de canales en MIRTH CONNECT **58**

C. Anexo: Plantilla de Archivo CDA.xsl **68**

D. Anexo: Instalación de Herramienta OSSEC IDS **69**

E. Anexo: Instalación de Herramienta Mod Security IPS **72**

F. Anexo: Informe Ejecutivo e Iso 27001 de Acunetix **74**

G. Anexo: LINK para la Descarga del PROTOTIPO DE FRAMEWORK MODSEC **75**

H. Anexo: Entrevista a Profesional en el área de sistemas de la entidad Promotora Médica las Américas **76**

Bibliografía **78**

Lista de Figuras

2-1. Estructura del Mensaje HL7 6

3-1. Metodología del Desarrollo del Proyecto 13
3-2. Ciclo de Identificar y Caracterizar 14
3-3. Motor de Envios Mirth Connect Interfaz Panel de Control 17
3-4. Motor de Envios Mirth Connect Interfaz de Conteo de envíos 17
3-5. Ciclo Desarrollo de Prototipo de Framework 19
3-6. Visor de Documentos Medical Record Viewer 20
3-7. Arquitectura de red simple para simulación 20
3-8. Escaneo de Wireshark a los mensajes enviados por Mirth connect 21
3-9. Detalles de CVE vulnerabilidades en HL7 CDA 21
3-10.Prototipo de Framework MODSEC 22
3-11.Ciclo Desarrollo de Evaluación del Framework 23

4-1. Datos HL7 sin cifrar recopilados de Wireshark en un sistema MITM atacante 27
4-2. Clasificación de incidentes en ataques 28
4-3. Modelo de red HL7 CDA R2 Punto a Punto 30
4-4. Modelo de red HL7 CDA R2 Motor de Interfaz 31
4-5. Estructura de Archivo CDA.xsl 32
4-6. Estructura de CDA en 2 partes 33
4-7. Estructura de CDA en 2 partes 34
4-8. Backbeach Software Medical Record Viewer 36
4-9. Carpeta de historiales Médicos CDA 37
4-10.Carpeta donde se realizará el envío 38
4-11.Desaparece el archivo CDA enviado 38
4-12.Carpeta Receptora DESTINO de los Archivos HL7 CDA R2 39
4-13.Confirmación en MIRTH CONNECT de envío EXITOSO 40
4-14.Contenido del Mensaje que se envió 40
4-15.Mensaje de envío exitoso a carpeta DESTINO DEMO DOCUMENTS 41
4-16.Cambios de método de conexión al enviar los CDA 41
4-17.Envío DESTINO por método POST en HTTP SENDER 42
4-18.IP ATACANTE bajo ambiente controlado 43
4-19.IP DESTINO bajo ambiente controlado 43
4-20.Envío Exitoso por método HTTP SENDER 44

4-21.Filtro de IP destino y puerto en WIRESHARK 44
4-22.Paquetes de WIRESHARK que contienen el mensaje enviado 44
4-23.Contenido de los Paquetes de WIRESHARK en ip destino 45
4-24.manejo predeterminado de un nonXMLBody CDA 46
4-25.Código JavaScript 46
4-26.Código incrustado en el IFRAME de alerta 46
4-27.Logotipo de la Herramienta OSSEC 47
4-28.Logotipo de la Herramienta Mod Security 47
4-29.Reinicio de Servicio apache2 48
4-30.Comando para Editar el archivo modsecurity.conf 49
4-31.Edición del Archivo modsecurity.conf 49
4-32.Reinicio de apache2 49
4-33.Envío del archivo HL7 CDA de forma correcta 50
4-34.Paquetes de Wireshark RST 50
4-35.Paquetes de Wireshark Out of Order 50
4-36.Paquete de Wireshark Sin Lectura 51
4-37.Framework de OSSEC y MOD SECURITY funcionando Exitosamente . . . 51
4-38.Página remota del Prototipo de Framework Caída 52
4-39.Resultado efectivo de la mitigación y detección del ataque 52

A-1. Instalación Motor MIRTH CONNECT 1 56
A-2. Instalación Motor MIRTH CONNECT 2 57
A-3. Instalación Motor MIRTH CONNECT 3 57

B-1. Cambio de Datos Personales Motor MIRTH CONNECT 58
B-2. Creación de canales en Motor MIRTH CONNECT 59
B-3. nombre del canal en Motor MIRTH CONNECT 60
B-4. Cambio de Formato de archivos en Motor MIRTH CONNECT 60
B-5. Tipo de Conexión e Intervalo de Envío en Motor MIRTH CONNECT 61
B-6. Eliminando Archivo enviado y conexión con directorio contenedor 61
B-7. Dirección de carpeta contenedora de CDA y filtro de archivos XML 62
B-8. Testeo de Conexión Correcta 62
B-9. Configuración Destino de archivos CDA (Entidad Hospitalaria Receptora) . 63
B-10 Editar transformación de la plantilla CDA.xsl 64
B-11 Abriendo plantilla de edición para que sea transformada en versión xml editable 64
B-12 Abriendo plantilla de edición para que sea transformada en versión xml editable 65
B-13 Plantilla editada terminada 65
B-14 Guardado de Canal y Receptor 66
B-15 Conexión completa de este canal entre Sistema 1 a Sistema 2 66

B-16 Opción Reploy All 67
B-17 Aceptación del canal en funcionamiento 67

C-1. Plantilla de Archivo CDA.xsl 68

E-1. Asegurar sí apache está instalado 72
E-2. Incluir línea de código 73

F-1. Informe Ejecutivo de Acunetix 74

G-1. PROTOTIPO DE FRAMEWORK MODSEC 75

Lista de Tablas

3-1. Identificación de los Incidentes bajo principales puntos de afectación en HL7 CDA R2 Sin Llenar 15
3-2. Selección de Motor de envío de mensajería sin Llenar 16
3-3. Selección de los mejores IDS e IPS 18

4-1. Identificación de los Incidentes bajo principales puntos de afectación en HL7 CDA R2 26
4-2. Motores de envío de mensajería HL7 29
4-3. Tabla de los mejores IDS e IPS 35

1. Introducción

Las entidades del sector salud, al necesitar los sistemas de información electrónico, dan a conocer las necesidades de implementar este sistema, bajo un estándar que apropie la interoperabilidad, la cual dará cabida a la interpretación semántica que sea legible al ojo humano y por lo tanto, sus datos e información serán compatibles con cualquier dispositivo, esperando un buen funcionamiento de envío de mensajes, superando expectativas de seguridad básicas.

En el tema de sistemas de información en salud, se ha abordado el estándar HL7, que es un paso para llegar a la interoperabilidad, "un estándar de interoperabilidad aprobado por ISO que proporciona un modelo de intercambio de documentos clínicos (por ejemplo, informes de alta o epicrisis y notas de evolución)" [1]. Este estándar ha sido importante para acercar al sector salud a la meta de una historia clínica electrónica compartida, HL7 trabaja directamente sobre la capa OSI 7 (CAPA de Aplicación). El estándar CDA pertenece a la organización HL7 que significa por sus siglas en inglés Clinical Document Architecture, Arquitectura de documentos clínicos, es un estándar diseñado para la representación y comunicación de documentos clínicos, estandariza exclusivamente la estructura y la semántica necesaria para el intercambio de los documentos clínicos. CDA fue realizado con él estándar HL7, es un estándar flexible en el sentido de que puede ser leído por el ojo humano o procesado por una máquina.

En Colombia, el estándar HL7 se ha venido implementando desde el año 2015, donde "el estándar fue aplicado a la plataforma OHEVS, el cual trata de una adaptación del estándar en hábitos y estilos de vida saludables. Esta es una plataforma interoperable, preparada para un observatorio de hábitos y estilos de vida saludables en población adulta y adulto mayor de poblaciones rurales de Colombia" [2], Dentro de los estándares complementarios de HL7 se encuentra el CDA R2 el cual se encarga de la estructura y semántica de los historiales o documentos médicos en su versión actual Release 2, pero por la tardía implementación del estándar HL7 en Colombia todavía no se ha pensado, por ahora, en una implementación del estándar CDA R2. Existen ya algunas implementaciones del estándar CDA en Norte y Sudamérica, en Europa y los países de Asia en la costa del Pacífico. El propósito que tiene el estándar es el de permitir un intercambio de información de Atención médica, a través de unas metodologías y servicios, ofreciendo la interoperabilidad en los sistemas de información de Salud de una manera ágil ya que la información transmitida por ese estándar es de carácter confidencial y es muy crítico si llegara a manos de delincuentes cibernéticos [2].

Debido a esto, se decide realizar una investigación cuyo objetivo general es: **"Desarrollar un Prototipo de Framework para la mitigación de riesgos en la confidencialidad de los datos del estándar HL7 CDA R2"**, y para el desarrollo y cumplimiento de este objetivo general se crean 4 objetivos específicos:

- Identificar y caracterizar los diferentes problemas de confidencialidad que se presentan en el estándar HL7 CDA R2.
- Determinar cuál es la herramienta idónea para la detección y prevención que afecte la confidencialidad en el estándar HL7 CDA R2.
- Implementar el prototipo de Framework de seguridad para preservar la confidencialidad de los datos en el estándar HL7 CDA R2.
- Evaluar la implementación del prototipo de Framework propuesto bajo pruebas de ambiente controlado para comprobar que los riesgos han sido mitigados.

Esta investigación necesariamente exige, en un primer momento, analizar las dificultades o problemas que se presentan a la hora de enviar las historias clínicas, salvaguardando los datos de los pacientes, datos que se envían por medio electrónico a través de internet, compartiendo cada vez más volúmenes extraordinarios de información, en condiciones tales como anchos de banda para poder comunicarse, preferentemente en una misma infraestructura tecnológica; es fundamental comprender que este estándar cuenta con seguridad básica [3], donde "se propone un modelo de control de acceso, el cual lleva cifrado parcial y utiliza una firma digital electrónica, pero que en el intento de unificar por medio de la interoperabilidad, el envío de los datos incide en el aumento de la inseguridad y aumenta las vulnerabilidades" [3]. No se trabajó con empresas reales, ya que las bases de datos son confidenciales y a la hora de realizar el análisis de vulnerabilidades y ataques, puede dañar o modificar los datos de los pacientes lo que llevaría a un daño a esta información, así mismo, es por esto que se trabajó bajo un ambiente controlado para salvaguardar la información y poder realizar los ataques sin problema alguno.
De allí nace la necesidad de identificar qué tipo de vulnerabilidades se encuentran en el estándar HL7 CDA R2, que afecten de manera directa o indirecta la confidencialidad de los datos de los pacientes, datos que son muy valiosos para diversos atacantes o delincuentes informáticos, porque pueden encontrar información muy sensible de los pacientes.
Cuando el estándar HL7 CDA R2 realiza la interoperabilidad entre equipos y entidades prestadoras de salud, se abren brechas de seguridad que deben ser tratadas en un carácter urgente por su nivel de alto riesgo en la confidencialidad de los datos, para ello se debe realizar un proceso de análisis de riesgos exhaustivo y análisis de vulnerabilidades para detectar a tiempo estas falencias y proceder rápidamente con la implementación de su seguridad. Con la creación de un prototipo de Framework para la seguridad en su confidencialidad de los

datos, se podría mitigar las vulnerabilidades críticas en la información de los pacientes del estándar Hospitalario HL7 CDA R2.
Se trabaja bajo una metodología única que conlleva a la unión de 3 ciclos PHVA para llegar a un objetivo general, el primer ciclo de PHVA está estructurado para solucionar y llegar al cumplimiento de los objetivos 1 y 2, el Segundo ciclo cumpliría el objetivo 3 y por último el ciclo Tercero daría por terminado el objetivo 4 de la tesis y así llegar al desarrollo total del proyecto, dando por terminado el objetivo general.

2. Marco Teórico y Estado del Arte

2.1. Marco teórico

2.1.1. Sistema de Información en Salud.

El sistema de Información en Salud recolecta e integra varios reportes médicos, con información primordial de carácter urgente y necesaria para el mejoramiento de los servicios en salud. Las entidades prestadoras de servicio en salud hoy en día se encuentran en la necesidad de acudir al intercambio de información en diferentes sistemas de registro electrónico, por ello se han creado mecanismos estandarizados para suplir con la necesidad de tener que enviar gran cantidad de datos de carácter urgente y confidencial por medio físico [4].
Al encontrar una necesidad de comunicar varias entidades de salud para transmitir información de forma segura se encuentran con un mecanismo estandarizado llamado HL7, para realizar esta labor se necesitan plataformas o frameworks interoperables que permitan realizar el envío de diferentes archivos y recepción de los mismos de forma legible para el humano es aquí donde entra en juego el concepto de interoperabilidad.
Uno de los principales retos en la actualidad, en investigaciones de seguridad informática, es comprender a profundidad el fenómeno de la interoperabilidad relacionada estrechamente con la seguridad en el estándar HL7 [5]. Es un tema relativamente nuevo cuando se trata de desarrollar una experiencia investigativa aplicado a la seguridad en un estándar, en este caso HL7. La Interoperabilidad se define como la capacidad que tienen los sistemas de información de comunicarse entre sí e intercambiar datos [6].

2.1.2. Interoperabilidad en Sistemas de Información en Salud.

Las investigaciones frente al estándar HL7 principalmente se basan en la parte operacional del estándar y se centran en el manejo de la interoperabilidad que se pide para el buen manejo de mensajería e historiales de salud con diferentes estructuras para que sean compatibles a la hora de necesitar estos datos [7][1][8][9].
La investigación de Bernal-Acevedo, habla acerca de los Sistemas de información en el sector salud en Colombia siendo el principal tema para entender como están distribuidos los sistemas de información en salud mediante un estudio descriptivo el cual utiliza metodologías cualitativas frente a la caracterización del sistema de información de otros países para tener así un conjunto de experiencias, debilidades y fortalezas en las cuales se pueden referenciar

para así posteriormente trabajar en un marco conceptual para el sistema de información de Colombia, trabajando conjuntamente con funcionarios del Instituto Nacional de Salud y del Ministerio de la Protección Social [7].
El estándar HL7 significa Salud Nivel 7 por sus siglas en inglés Health Level 7, la cual es una organización internacional no gubernamental y sin fines de lucro que se encarga de generar estándares de interoperabilidad para todo tipo de informática médica, certificada por ANSI (American National Standards Institute), cuenta con más de 35 países a nivel mundial con su afiliación creando nuevas especificaciones y estándares. Es así como HL7, es una organización con un conjunto de estándares que dictan el formato en el que se puede intercambiar información electrónicamente entre los diferentes sistemas informáticos de los prestadores de servicios de salud.
El estándar HL7 sirve para realizar intercambio de información del paciente como su nombre, su edad, si tiene hermanos, son datos básicos que se transmiten de una entidad a otra, un ejemplo de la estructura de un mensaje en el estándar HL7 se puede apreciar en la Figura **2-1**

```
MSH|^~\&|SENDING_APPLICATION|SENDING_FACILITY|RECEIVING_APPLICATION|RECEIVING_F
ACILITY|20110613083617||ADT^A01|934576120110613083617|P|2.3||||
EVN|A01|20110613083617|||
PID|1||135769||MOUSE^MICKEY^||19281118|M|||123 Main St.^^Lake Buena Vista^FL
^32830||(407)939-5555^^^ohtoodles@notdisney.com|||||1719|999999999|||
|MOUSETOWN|||||||||||||||
NK1|1|MOUSE^MINNIE|WIFE||||||NK
PV1|1|O|||||^^^^^^^^|^^^^^^^^
AL1|1||^Penicillin||Anaphylactic shock
AL1|2||^Cat dander||Skin rash
```

Figura 2-1.: Estructura del Mensaje HL7, **Fuente:** [10]

2.1.3. Estándar HL7 CDA R2 (Release 2)

Es un estándar de marcado de documentos que especifica la estructura y la semántica de los "documentos clínicos"con el fin de intercambiar entre los proveedores de atención médica y los pacientes.
Define un documento clínico que tiene las siguientes seis características:

1. "persistencia

2. administración

3. potencial de autenticación
4. contexto
5. integridad
6. legibilidad humana. [11]"

Las investigaciones en el mundo se realizaron con el propósito de dotar interoperabilidad con estándares internacionales como HL7 CDA ya que este estándar tiene muchas herramientas para compartir información en salud. En el ámbito de la salud conlleva a un intercambio alto de información y requiere un flujo continuo de dicha información, para poder lograr este objetivo se requiere asegurar dicha interoperabilidad de los sistemas de información que dan soporte al proceso asistencial, este intercambio de información puede realizarse utilizando estándares como HL7 CDA. La Organización Panamericana de la Salud presenta un reporte con la finalidad de describir un marco conceptual sobre la interoperabilidad y estándares en salud el cual realiza una revisión de la literatura en la implementación de estándares para lograr la interoperabilidad en los países de Latinoamérica y el Caribe [12].
Estos estándares son muy buenos para generar la interoperabilidad entre las entidades prestadoras de servicio en salud, es por eso que dicho estándar se encuentra implementado en 35 países, dada la relevancia que tiene el HL7 CDA en las diferentes latitudes, nace la necesidad de implantar características de seguridad y realizar un seguimiento paso a paso de esta, obteniendo de este seguimiento el material necesario para desarrollar análisis de vulnerabilidades en el sistema.
Cuando el estándar HL7 CDA R2 realiza la interoperabilidad entre equipos y entidades prestadoras de salud se abren brechas de seguridad que deben ser tratadas en un carácter urgente por su nivel de alto riesgo en la confidencialidad de los datos, para ello se debe realizar un proceso de análisis de vulnerabilidades exhaustivo para detectar a tiempo estas falencias y proceder rápidamente con la implementación de su seguridad.

2.1.4. Análisis de Vulnerabilidades

Es un proceso en el cual las organizaciones determinan los niveles de exposición de perdida de elementos ante una amenaza, por medio de este proceso se determina que tan crítica es la seguridad y se detecta que tipo de falencias tiene para así crear un informe que permita la mitigación de estas vulnerabilidades. Existen herramientas que pueden simplificar el tiempo y realizar este proceso mecánico de una manera rápida y verídica ejemplo, (Acunetix, Nessus…etc.)[13].

2.1.5. Modulo OSSEC

Es "un sistema de detección de intrusiones gratuito, de código abierto basado en host. Realiza análisis de registros, verificación de integridad, monitoreo del registro de Windows, detección

de rootkits, alertas basadas en el tiempo y respuesta activa, es utilizado en sistemas de capa de aplicación (capa 7) el cual es muy adecuado para la detección y protección del estándar HL7".[14].

2.1.6. Modulo ModSecurity

ModSecurity es considerado como un módulo, en su principio se creó para Apache pero ya está disponible para Microsoft IIS y NginX, el cual permite mantener la seguridad de un servidor web y ayuda a protegerlo de posibles ataques, funciona bajo expresiones regulares y reglas que se pueden personalizar a conveniencia de la empresa u organización que lo este implementando, este producto fue diseñado por Breach Security y está disponible como software libre bajo la licencia GNU [15]. Trabaja protegiendo ataques como los estipulados por OWASP en su top 10 y su funcionalidad es de un IPS (Sistema de prevención de Intrusos) el cual ayuda a prevenir los ataques mal intencionados:

- **"inyección.**
- **Autenticación rota.**
- **Exposición de datos sensibles.**
- **Entidades externas XML (XXE).**
- **Control de acceso roto.**
- **Mala configuración de seguridad.**
- **Cross-Site Scripting XSS.**
- **Deserialización insegura.**
- **Uso de componentes con vulnerabilidades conocidas.**
- **Insuficiente registro y monitoreo."[16]**

2.1.7. Mirth Connect

Este es un motor de integración hl7 comúnmente llamado middleware, el cual facilita el intercambio de mensajes, enrutamiento, extracción y transformación de ellos, administra, despliega y monitoriza las interfaces de HL7 por medio de canales que se crean como conectores de transporte necesarios para guiar a los mensajes. Esta es una Herramienta de integración especializada en SALUD con unos conversores de estándares de la mensajería en salud de HL7, de Licenciamiento OPEN SOURCE bajo la licencia publica MPL (Mozilla

Public License 1.1) con soporte por medio de la comunidad de usuarios y su licencia comercial con el soporte directo por el staff de Mirth Corporation. Ideal para grandes proyectos de interoperabilidad en salud, cuenta con un programa de capacitación y certificación por medio de la empresa NEXTGEN[17][18].

2.1.8. WHIRESHARK

Es una herramienta especializada en analizar protocolos de red, es Open Source y está disponible en diversas plataformas. ''Excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red'' [19]. Sirve también como herramienta de análisis de vulnerabilidades bajo una técnica llamada Sniffer el cual se encarga de capturar paquetes enviados por la red.

2.1.9. Man in the Middle (MitM)

Es uno de los ataques más habituales, consiste en interceptar la comunicación que existe con el usuario, en español esta técnica hacking significa hombre en el medio, ataque muy efectivo para la fuga de información y obtención de datos de carácter confidencial. Existen muchas herramientas con las cuales se puede hacer un MitM pero las más efectivas siempre serán los Sniffer ya que a la hora de capturar los paquetes estos pueden ser modificados y ser nuevamente enviados por el mismo segmento de red y así saltar alertas de firewalls[20].

2.1.10. cross site scripting (xss)

El XSS se considera un ataque de inyección de código malicioso que se realiza a los aplicativos web colocando información en forma de hipervínculo y así redirigir al usuario hacia otra pagina, se puede explotar de dos formas, de manera reflejada y de forma almacenada. La forma reflejada consiste en la modificación de valores en la aplicación web para así usar variables entre dos paginas y la almacenada consiste en inyectar código malicioso HTML para que los usuarios ingresen al aplicativo web modificado[21].

2.1.11. Open Web Application Security Project (OWASP)

Es una organización abierta sin ánimo de lucro el cual maneja el aseguramiento en aplicativos web, esta organización nació en septiembre del 2001, busca constantemente fomentar el desarrollo seguro en seguridad web, mantiene las buenas prácticas de seguridad y metodologías para el desarrollo web. El top 10 es un documento donde se enumeran los problemas o riesgos más frecuentes que se encuentran en aplicaciones móviles y aplicaciones web según el tipo de aplicación tiene su top,la versión más reciente es la publicada en 2017[22].

2.1.12. ¿Qué es un IDS y un IPS?

IDS, el significado de estas siglas es Sistema de Detección de Intrusiones, este es un sistema que actúa para la posterior protección de una infraestructura, detecta los ataques como acceso no autorizado en la red, llevando a cabo un análisis en tiempo real de las conexiones y así determinar si se está produciendo o se va a producir un incidente, puede detectar si es un ataque directo o un falso positivo, este se recomienda trabajarlo en conjunto con un firewall, ya que no tiene la funcionalidad de bloquear un ataque[23].
IPS por sus siglas Sistema de prevención de intrusos, es un sistema de prevención y protección el cual se defiende de intrusiones y tiene la habilidad de bloquear los ataques sin importar el protocolo de transporte que se esté utilizando, controla los accesos de usuarios ilegítimos, está basado en firmas, políticas y anomalías, protege de forma proactiva la red[23].
En si son sistemas de detección y de prevención de ataques expuestos en herramientas de forma virtual y físicas que ayudaran a proteger los sistemas de información de una empresa o entidad.

2.1.13. ¿Qué es CVE?

La CVE por sus siglas Common Vulnerabilities and Exposures es una lista de información en la cual se encuentran registradas las vulnerabilidades de seguridad informática más conocidas e importantes, y se encuentran identificadas por un código único para tener una búsqueda más rápida y organizada con el fin de facilitar el intercambio de información entre diferentes bases de datos de vulnerabilidades existentes[24].

2.1.14. Framework

Este es un espacio de trabajo el cual es encargado de la seguridad de las aplicaciones, permite desarrollar la seguridad de las aplicaciones de forma generalizada y sin la necesidad de desarrollar módulos de seguridad individuales para cada aplicación[25].

2.2. Estado del Arte

Los estudios realizados a nivel mundial sobre el estándar HL7 son bastantes, ya que conllevan a tratar de implementar el estándar en todos los países para trabajar con una interoperabilidad funcional, sin embargo al hablar del estándar como HL7 CDA R2 se empiezan a delimitar los estudios, esto a costa de que no todos los países utilizan este estándar hospitalario para la transmisión de mensajes y así proceder a la formación de estos historiales clínicos electrónicos (CDA)[26][27][17], los estudios encontrados se basan más en tratar de encontrar una interoperabilidad entre sistemas y plataformas para su fácil manejo y de interfaz amigable para el usuario final [28][29][30], a pesar de este contratiempo se pudo llegar a algunos estudios los cuales hablan muy superficialmente de su seguridad y que para

posteriores investigaciones es un reto llegar a obtener esta interoperabilidad sin afectar los problemas de seguridad[31]. Shahid Munir Shah menciona que los datos contenidos en los registros electrónicos de salud (EHR) no solo se utilizan para la atención primaria de los pacientes, sino también para diversos fines secundarios como son las auditorías clínicas o una vigilancia automatizada, incluso para investigación clínica, no hay un consentimiento para el uso de estos datos, aunque algunos si lo tengan crean problemas de confidencialidad a quedar expuestos, La fuga de datos puede causar pérdidas financieras o un individuo puede encontrar un boicot social si su condición médica se expone en público, dicho estudio tiene como objetivo resaltar cómo estos usos secundarios afectan la privacidad de los pacientes[27].

Se realizó un estudio y se estableció un consenso sobre los principales criterios para el desarrollo de documentos CDA en España, con el fin de ayudar a implementar y desarrollar documentos clínicos en el estándar CDA, a pesar que el estudio desglosa muy bien el funcionamiento del estándar HL7 CDA, cumpliendo con los niveles de interoperabilidad principales como son el sintáctico y semántico, los cuales permiten el buen manejo de envío de documentos en salud, no se logra apreciar una profundidad en el manejo de la seguridad de dichos documentos, por cuanto simplemente, se nombra la confidencialidad de los datos como un tema a tratar en segundo plano [1];
este estudio se centra más en la estructura y código del estándar HL7 CDA. En su estudio SALUD E INTEROPERABILIDAD EN COLOMBIA, no toma como necesidad la seguridad del estándar en su confidencialidad, solo hace énfasis en los avances de interoperabilidad en documentos del sector Salud, retoman proyectos implementados en otros países para tener referencias positivas y así, poder continuar con la implementación en Colombia del estándar HL7, que se enfoca en la evolución de los sistemas de información [9].

En el estudio realizado por Mandirola, Cesar Moreno, Brieux F, Rosa y Ricardo Herrero, le da mucha más importancia a la seguridad, haciendo referencia a las deficiencias que tiene el estándar y tomando como principal activo la seguridad de la información de los pacientes, dando algunas definiciones de la seguridad de la información y de los tres conceptos básicos: Disponibilidad, Confidencialidad e Integridad [32], tomando así como los dos más importantes, que son la Confidencialidad Vs Disponibilidad ya que son interrelacionados porque "Toda medida para facilitar la disponibilidad de los datos penaliza su confidencialidad, y viceversa. Se debe garantizar un equilibrio razonable entre ambos extremos" [32]. Las soluciones previstas a estos problemas las dan por medio de leyes vigentes como (ley 26529 y 25.326) en los derechos de los ciudadanos la cual incluyen la protección de la salud y de la intimidad, obligan a las instituciones de salud a adoptar las medidas de seguridad necesarias para su garantía, (Ley 26.529) donde se habla de la información clínica la cual debe estar protegida y disponible, pero solo se nombra las leyes existentes.
Es por eso que en Colombia se encuentran en planes de proceso de implementación de historiales médicos electrónicos, pero no garantizan la seguridad en la confidencialidad, por

lo tanto, en la Universidad de San Buenaventura en la ciudad de Cali, con estudiantes del programa de desarrollo de ingeniería de software, realizan un análisis de varias brechas existentes en las normas Colombianas con respecto a la seguridad de los historiales médicos electrónicos y proponen un conjunto de directrices para gestionar y custodiar los registros electrónicos, pero a pesar de contribuir con varias directrices a nivel internacional no hace referencia alguna del estándar interoperable HL7 CDA el cual permite el envío de historiales médicos electrónicos entre entidades bajo estructura de plantillas y sus brechas de seguridad encontradas en la capa 7 [8].

3. Metodología

En la Figura **3-1** se puede apreciar el flujo de la metodología para poder llegar a concluir el proyecto y dar respuesta al objetivo general.

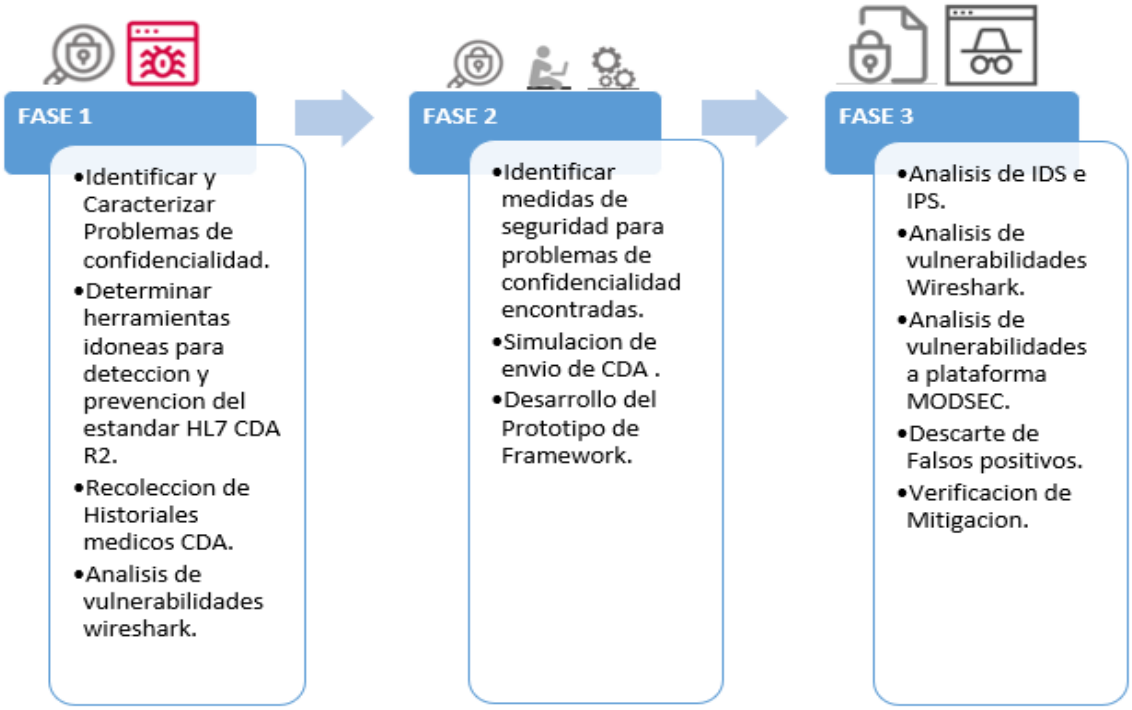


Figura 3-1.: Metodología del Desarrollo del Proyecto, **Fuente:** Elaboración Propia

Se trabaja bajo una organización de ciclos PHVA divididos de la siguiente manera:

- **Primera fase**, comprende el desarrollo de Actividades para los objetivos 1 y 2.
- **Segunda fase**, comprende el desarrollo de Actividades para el objetivo 3
- **Tercera fase**, comprende el desarrollo de Actividades para el objetivo 4

Estas fases son de terminación consecutiva, al terminar la primera fase de un ciclo PHVA se da comienzo a las actividades de la siguiente fase.

3.1. Fase 1, Identificar problemas y Determinar Herramientas.

En la Figura **3-2** se muestra el ciclo PHVA.

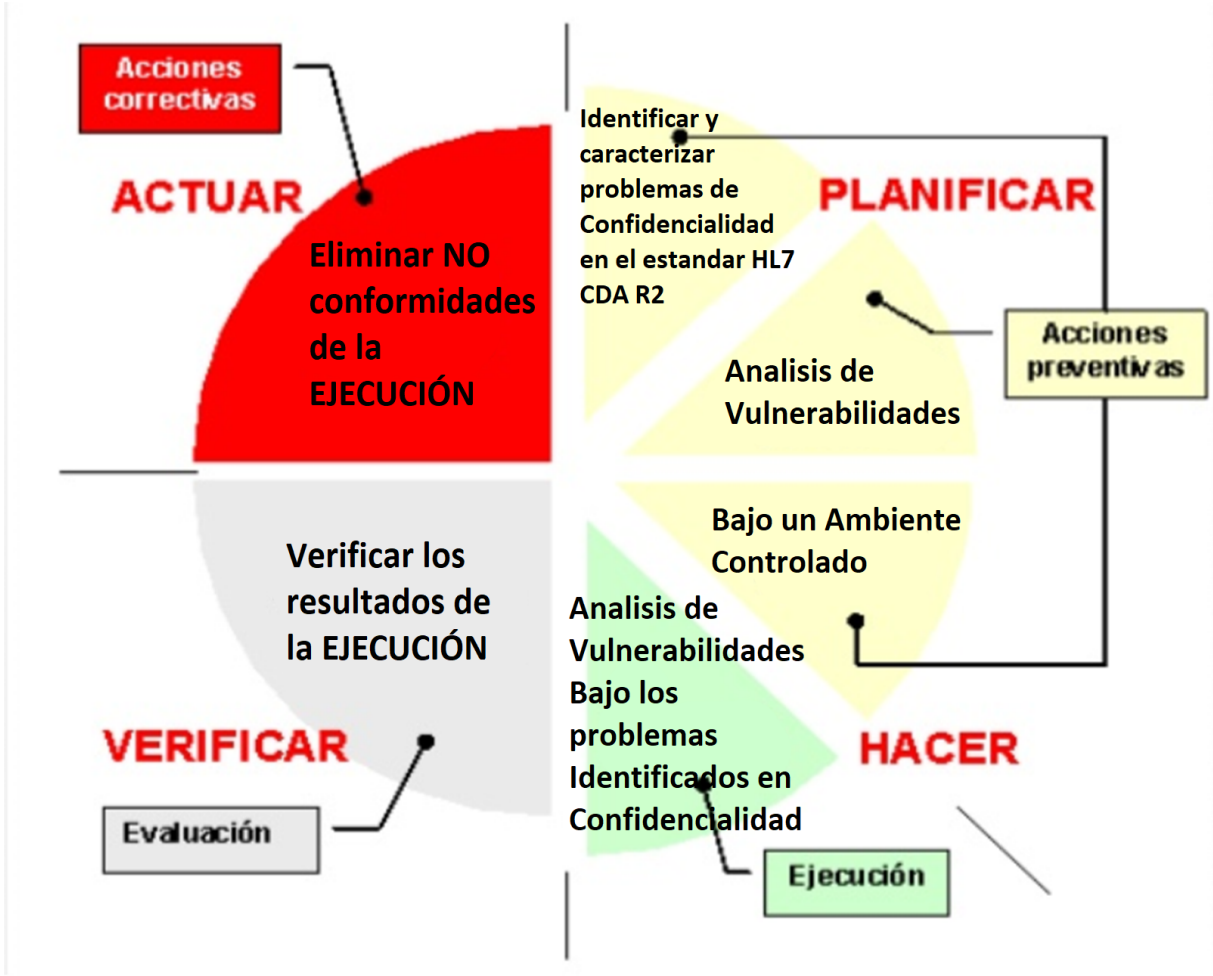


Figura 3-2.: Ciclo de Identificar y Caracterizar, **Fuente:** Elaboración Propia

En primera instancia Se aplicó una entrevista a un profesional capacitado que trabaja con HL7 en la promotora médica Las Américas de la Ciudad de Medellín, para tener el punto de vista de la experiencia de trabajar con el estándar y verificar que tipo de fallas ha tenido a la hora de trabajar con HL7 y saber si ha sufrido ataques comprometedores, que tipos de ataques y como los ha mitigado, así se obtuvo un punto de vista más amplio en un ambiente de trabajo real con el estándar HL7 CDA R2, ver Anexo H.
Se consultó los principales incidentes del estándar HL7 CDA R2 tomando como referencia anteriores estudios de seguridad[33][34][35][36], encontraron 6 incidentes que afectan el estándar HL7, estas investigaciones se realizaron bajo un análisis de vulnerabilidades, enviando

mensajería HL7 por la red, aunque los estudios solo se centraron en la mensajería HL7, tiene componentes importantes para estudiarlos con el estándar HL7 CDA R2, para así poder empezar con una base investigativa en este estándar. Principalmente en un análisis realizado por la empresa SMART donde se detalla los sectores de las vulnerabilidades encontradas las cuales fueron reportadas al CVE Vulnerability en el año 2014, dando a conocer los fallos principales del estándar HL7 CDA R2 (Inyección de código Malicioso(XSS) y Obtención de Información (MitM)) [37].
En la Tabla **3-1** se correlacionan las vulnerabilidades con las principales características necesarias para el estudio, ya que no todas estas vulnerabilidades actúan en la confidencialidad de los datos. Se parte desde los Problemas:

- Ejecución remota de código
- Comunicaciones de Texto Claro
- Interceptación de datos
- Pérdida de Autenticación y Gestión de Sesiones
- Fuga de información
- alteración de los datos

Y se termina con las características a tener en cuenta para el estudio, en este caso la confidencialidad, que pertenezcan al TOP 10 de OWASP y lógicamente que afecten el estándar HL7 CDA R2. Estas características se tomaron por la importancia de cada una de ellas en el estudio, la confidencialidad porque es la base de la seguridad en documentos clínicos y debe ser parte fundamental del estudio, OWASP por la importancia del top 10 en vulnerabilidades ya que los IPS trabajan con reglas predefinidas en este top y HL7 CDA R2 porque es el estándar a trabajar en el proyecto.

PROBLEMAS	CONFIDENCIALIDAD	OWASP	HL7 CDA R2
Ejecución remota de código			
Comunicaciones de Texto Claro			
interceptación de datos			
Pérdida de Autenticación y Gestión de Sesiones.			
fuga de información			
alteración de los datos.			

Tabla 3-1.: Identificación de los Incidentes bajo principales puntos de afectación en HL7 CDA R2 Sin Llenar, **Fuente:** Elaboración Propia

Después de identificar los principales Incidentes se pasó a caracterizarlos identificarlos y separarlos para determinar cuales son los ataques que afectan la confidencialidad del estándar,

esto se realizó con la ayuda de la Tabla **3-1**.
Se procedió a diseñar una red de envío de historiales médicos bajo el estándar HL7 CDA entre 2 entidades, se diseñó un envío simple entre 2 hospitales ficticios en un entorno controlado llamado Sistema 1 y Sistema 2 , para ello se procedió a escoger un motor de envío de mensajería en HL7. Para escoger este motor se realizó una matriz selectiva entre varios motores de HL7, ver Tabla **3-2**, basado en características fundamentales a tener en cuenta, las cuales afectaran el manejo y comportamiento del motor. Como el fácil manejo en el sistema operativo implementado, Open Source, facilidad de envío de mensajería HL7, rápida traducción e interpretación de la mensajería HL7 CDA R2, estas características son fundamentales para el buen funcionamiento y fácil implementación del motor, se escogieron estas características con el fin de tener el motor más adecuado a esta investigación y con pautas claves que son necesarias para poder implementarlas como que sea compatible con CDA y con XML y ante todo sin perder una de las características principales por lo que la empresa HL7 International ha estado investigando la INTEROPERABILIDAD.

Motores de Envío mensajería HL7	**Interoperabilidad**	**Open Source**	**Fácil Implementación**	**Múltiples Formatos de Mensajería**	**Compatible con CDA**	**Compatible con XML**	**Seguridad Básica**
Iguana							
DICOM PACS							
E-RESETAR							
HL7 Soup							
Quick view HL7							
Smarth HL7 Tools Bundle							
Mirth Connect							

Tabla 3-2.: Selección de Motor de envío de mensajería sin Llenar, **Fuente:** Elaboración Propia

el motor escogido fue Mirth Connect, bajo unos requerimientos mínimos de instalación en servidores Linux, pero se puede instalar en más S.O, solo se necesita de una maquina con estos requerimientos mínimos:

- Java 6 o superior.
- Al menos 100 MB de espacio libre en disco.
- S.O Windows (XP, Vista, 7, Server; 32 o 64 bits).
- S.O Linux (Kernel 2.4 o superior).
- Mac OS X (10X).
- Linux.

Este motor cuenta con interfaz de panel de control donde se realizan las configuraciones de los canales de envío y recepción de los documentos, ver Figura **3-3**, y como aplicación

web cuenta con una interfaz de monitoreo donde se aprecia estadísticamente el envío de los documentos ver Figura **3-4**.

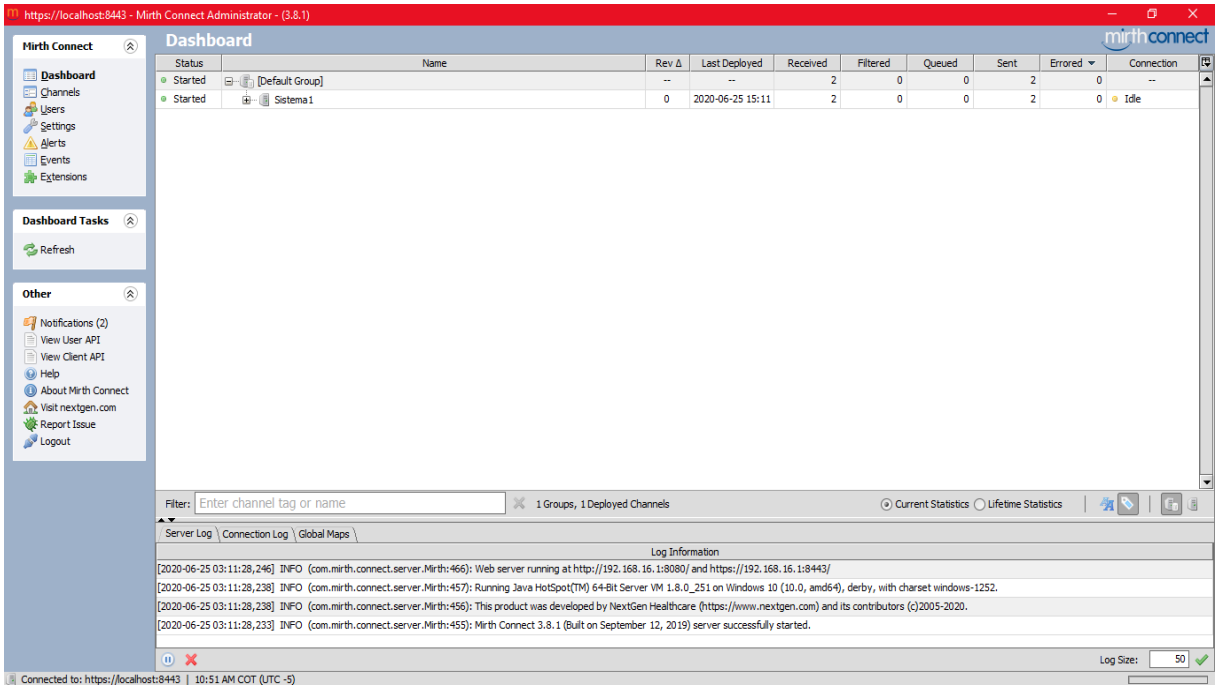


Figura 3-3.: Motor de Envios Mirth Connect Interfaz Panel de Control, **Fuente:** Elaboración Mirth Connect [17][18]

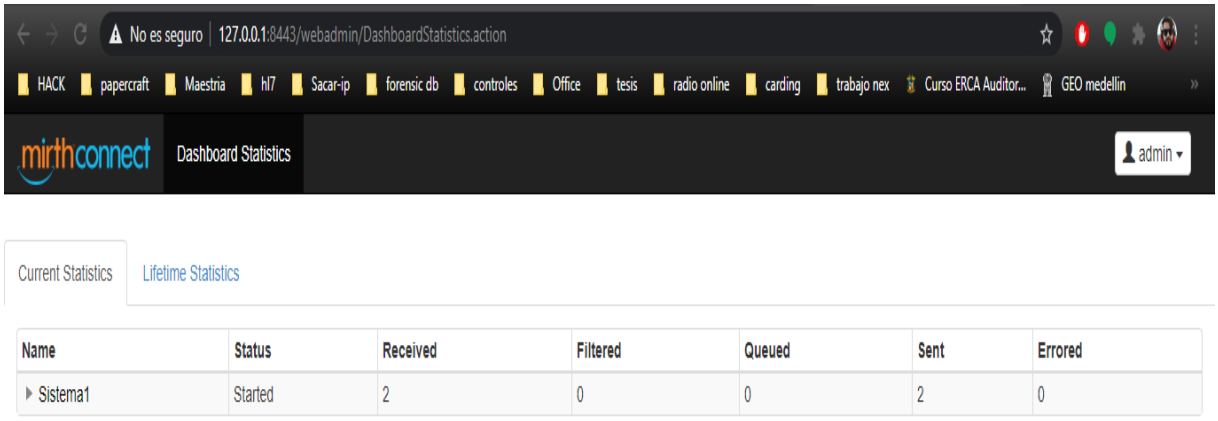


Figura 3-4.: Motor de Envios Mirth Connect Interfaz de Conteo de envíos, **Fuente:** Elaboración Mirth Connect [17][18]

Se determinaron cuales son las mejores e idóneas herramientas para la detección y protección

de vulnerabilidades, para ello se lleva a cabo una serie de pruebas bajo ambientes controlados, con máquinas virtuales en el programa VMware Workstation Pro con características específicas de 30 gb en disco duro y 4 en ram, el análisis de estas herramientas se basó en un cuadro correlacional de características fundamentales que deben tener estas herramientas o módulos como lo son el fácil manejo a la hora de su instalación el cual tenga buenas referencias y completas de su implementación para el sistema operativo escogido, que estén diseñadas para la capa de Aplicación (Capa 7) y que sean Open Source, estas características se tomaron en cuenta por la importancia que tiene trabajar bajo la capa 7, bajo herramientas que no tengan un costo pero que trabajen de forma efectiva y para la instalación que sea de fácil implementación y descarga, ver Tabla **3-3**.

Herramientas	**Open Source**	**Descarga Fácil**	**Fácil Implementación**	**Capa 7**
Fortinet (IPS)				
Mod Security (IPS)				
Nozomi SCADA Guardian (IDS)				
CISCO NGIPS (IPS)				
Corelight y Zeek (IDS)				
OSSEC (IDS)				
Fidelis Network (IPS)				
FireEye (IDS)-(IPS)				
SNORT (IDS)				

Tabla 3-3.: Selección de los mejores IDS e IPS, **Fuente:** Elaboración Propia

Estas Herramientas se probaron y analizaron bajo las características mencionadas en la Tabla **3-3**, obteniendo un resultado favorable a la hora de detectar y prevenir, las herramientas escogidas fueron el HIDS (Host Intrusion Detection System) OSSEC en su versión open source en conjunto de un IPS (intrusion prevention system) MOD SECURITY 3,0, herramientas que cumplieron en su totalidad con las características asignadas.

3.2. Fase 2, Desarrollo del prototipo de Framework.

En la Figura **3-5** se muestra el ciclo PHVA.

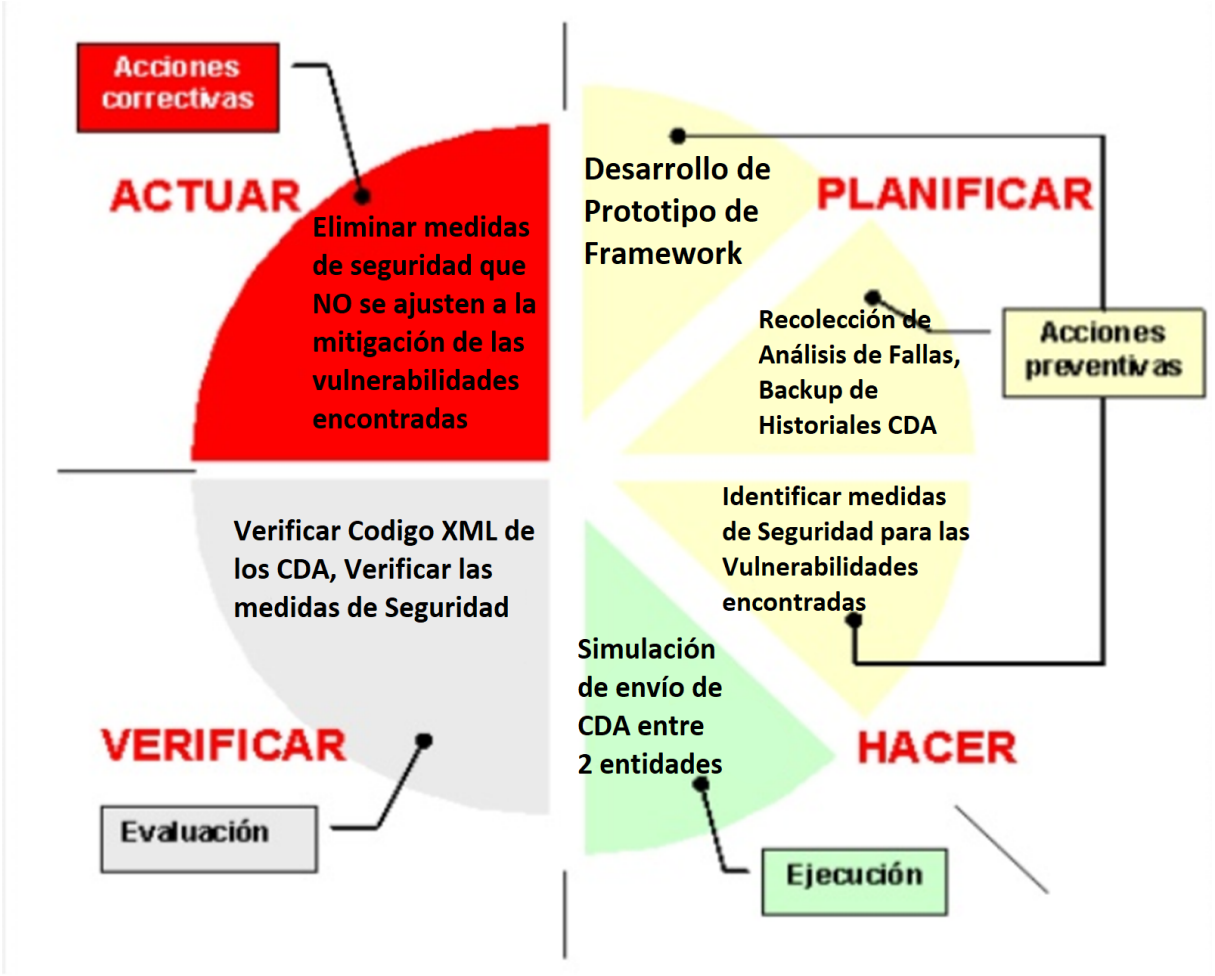


Figura 3-5.: Ciclo Desarrollo de Prototipo de Framework, **Fuente:** Elaboración Propia

En la siguiente fase se desarrolló un prototipo de Framework, con el fin de mitigar las vulnerabilidades. Primero que todo se consiguió los historiales médicos en XML para poder ser enviados por el estándar HL7 CDA R2, se encuentran en Idioma Ingles ya que la empresa que proporciono estos historiales es la empresa HL7 de Estados Unidos, para poder visualizar estos historiales médicos se tuvo que contar con un visor web de documentos CDA el cual fue proporcionado por la persona ganadora del concurso HL7 CDA View de la empresa HL7 Estados Unidos, esta herramienta es open source y ayuda a visualizar los documentos que son enviados bajo una plantilla XML, es así como se obtuvo un mejor manejo de interfaz amigable y legible para las personas, ver Figura **3-7**.

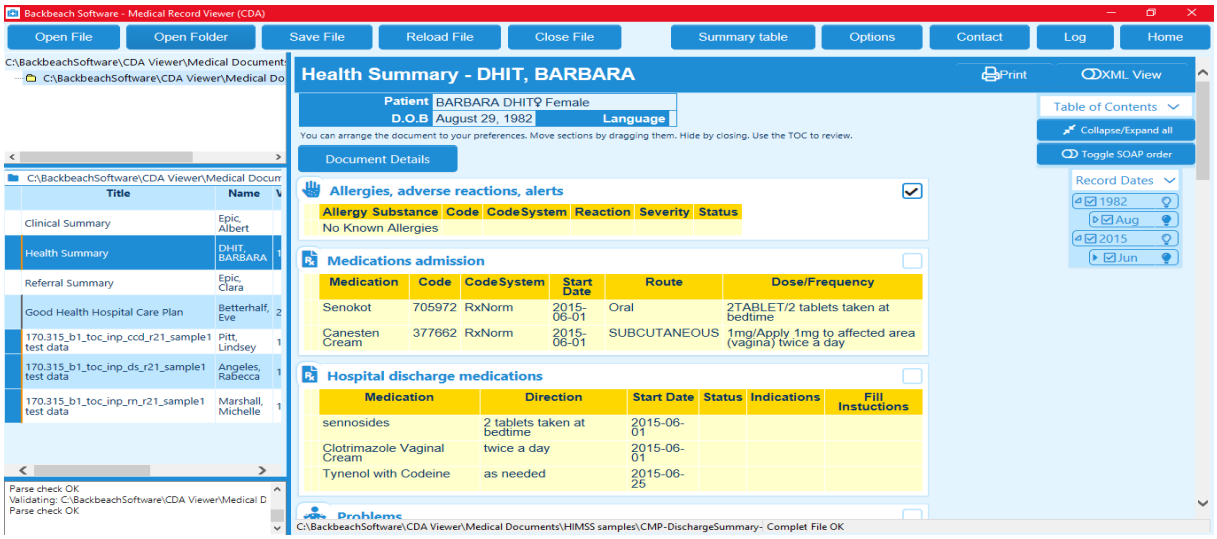
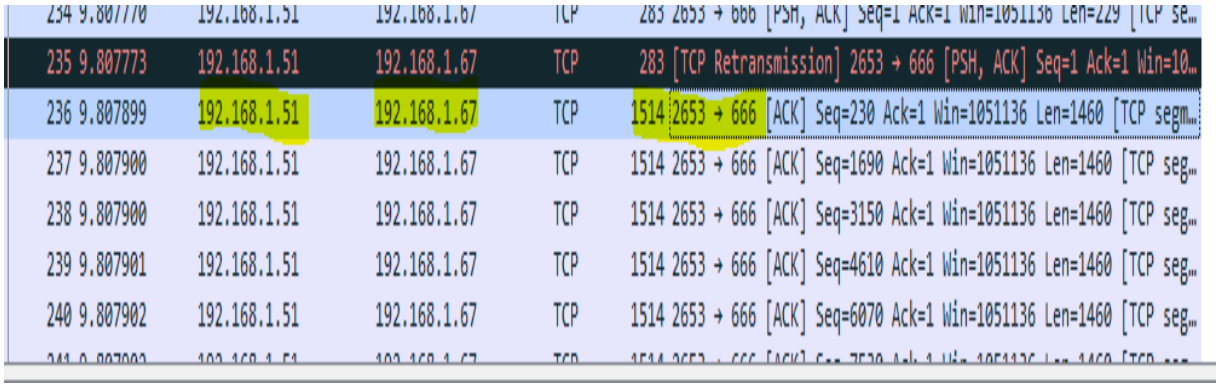


Figura 3-6.: Visor de Documentos Medical Record Viewer, **Fuente:** Elaboración backbeach software[38]

Se realiza la Simulación de envío de los historiales médicos entre 2 entidades de salud ficticias creadas con una arquitectura de red simulada simple, ver Figura **3-7**, para así proceder a la detección de vulnerabilidades por medio de la herramienta Wireshark como analizador de vulnerabilidades vía red, en la Figura **3-8** se puede apreciar el análisis de wireshark a un paquete enviado por el motor Mirth connect.

Figura 3-7.: Arquitectura de red simple para simulación, **Fuente:** Elaboración Propia



234 9.807770	192.168.1.51	192.168.1.67	TCP	283 2653 → 666 [PSH, ACK] Seq=1 Ack=1 Win=1051136 Len=229 [TCP Se…	
235 9.807773	192.168.1.51	192.168.1.67	TCP	283 [TCP Retransmission] 2653 → 666 [PSH, ACK] Seq=1 Ack=1 Win=10…	
236 9.807899	192.168.1.51	192.168.1.67	TCP	1514 2653 → 666 [ACK] Seq=230 Ack=1 Win=1051136 Len=1460 [TCP segm…	
237 9.807900	192.168.1.51	192.168.1.67	TCP	1514 2653 → 666 [ACK] Seq=1690 Ack=1 Win=1051136 Len=1460 [TCP seg…	
238 9.807900	192.168.1.51	192.168.1.67	TCP	1514 2653 → 666 [ACK] Seq=3150 Ack=1 Win=1051136 Len=1460 [TCP seg…	
239 9.807901	192.168.1.51	192.168.1.67	TCP	1514 2653 → 666 [ACK] Seq=4610 Ack=1 Win=1051136 Len=1460 [TCP seg…	
240 9.807902	192.168.1.51	192.168.1.67	TCP	1514 2653 → 666 [ACK] Seq=6070 Ack=1 Win=1051136 Len=1460 [TCP seg…	

Figura 3-8.: [Escaneo de Wireshark a los mensajes enviados por Mirth connect, **Fuente:** Elaboración Propia

Se consultaron y verificaron las medidas de seguridad para los Incidentes encontrados, mediante la implementación y acoplamiento del IDS e IPS los cuales se encargaron de detectar los ataques y prevenirlos con sus reglas ya predefinidas bajo el top de OWASP, es así como se pudo conocer el medio de mitigación de ellas aplicando estos controles, se estudiaron las medidas de seguridad para el diseño del prototipo de framework y así llegar a integrar la vista de una interfaz que sea más completa de las vulnerabilidades detectadas en tiempo real en conjunto con la base de datos de las CVE Vulnerability la cual se encuentra en la Figura **3-9**, para proporcionar el SCORE de la vulnerabilidad detectada por estas dos aplicaciones escogidos para la detección y contención.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2014-5452	79		XSS	2014-09-02	2016-12-21	4.3	None	Remote	Medium	Not required	None	Partial	None
CDA.xsl in HL7 C-CDA 1.1 and earlier does not anticipate the possibility of invalid C-CDA documents with crafted XML attributes, which allows remote attackers to conduct XSS attacks via a document containing a table that is improperly handled during unrestricted xsl:copy operations.														
2	CVE-2014-3862	200		+Info	2014-09-02	2014-09-02	4.3	None	Remote	Medium	Not required	Partial	None	None
CDA.xsl in HL7 C-CDA 1.1 and earlier allows remote attackers to discover potentially sensitive URLs via a crafted reference element that triggers creation of an IMG element with an arbitrary URL in its SRC attribute, leading to information disclosure in a Referer log.														
3	CVE-2014-3861	79		XSS	2014-09-02	2014-09-02	4.3	None	Remote	Medium	Not required	None	Partial	None
Cross-site scripting (XSS) vulnerability in CDA.xsl in HL7 C-CDA 1.1 and earlier allows remote attackers to inject arbitrary web script or HTML via a crafted reference element within a nonXMLBody element.														

Total number of vulnerabilities : 3 Page : 1 (This Page)

Figura 3-9.: Detalles de CVE vulnerabilidades en HL7 CDA, **Fuente:** Base de Datos CVE

Cabe resaltar que al no encontrar estudios previos de una implementación y acoplamiento de un IDS y un IPS Open Source, para esta investigación se realizó la instalación y configuración de estos dos módulos, ver Anexo D y Anexo E, para que trabajen de forma unificada MOD SECURITY y OSSEC dando el nombre del prototipo de FRAMEWORK MODSEC, ver Anexo G, la investigación se basó en qué tan estable trabajan los módulos en conjunto, haciendo pruebas de ensayo y error en un servidor Linux y basando esta integración en la

conferencia impartida en el año 2019 en el instituto tecnológico metropolitano (ITM) en el 15 festival de instalación de software libre (FLISOL), conferencia impartida por Paulo Diaz, Irving Solsol y Luis Flores bajo el título ¿Te sientes Protegido en la Red? donde se mostró el funcionamiento de los dos módulos bajo un mismo servidor, funcionando de manera correcta sin interrupciones ni errores, es así como se lo complementó con el score de la CVE como se ve en el prototipo de la Figura **3-10**. El funcionamiento de estas herramientas se conoció en el transcurso de la Maestría de Seguridad Informática en 2 materias diferentes impartidas por los Mg Milton Mateus y el Mg Andrés Gómez, al ver que son módulos muy eficientes y que cumplen con su labor de IDS e IPS, se procede a realizar pruebas de compatibilidad, en la instalación se observó comandos parecidos en los dos módulos, como en la instalación del apache2 y el mysql server, es por esto que se activaron los dos módulos al mismo tiempo, bajo una configuración de reglas sin dar problema alguno en su funcionamiento.

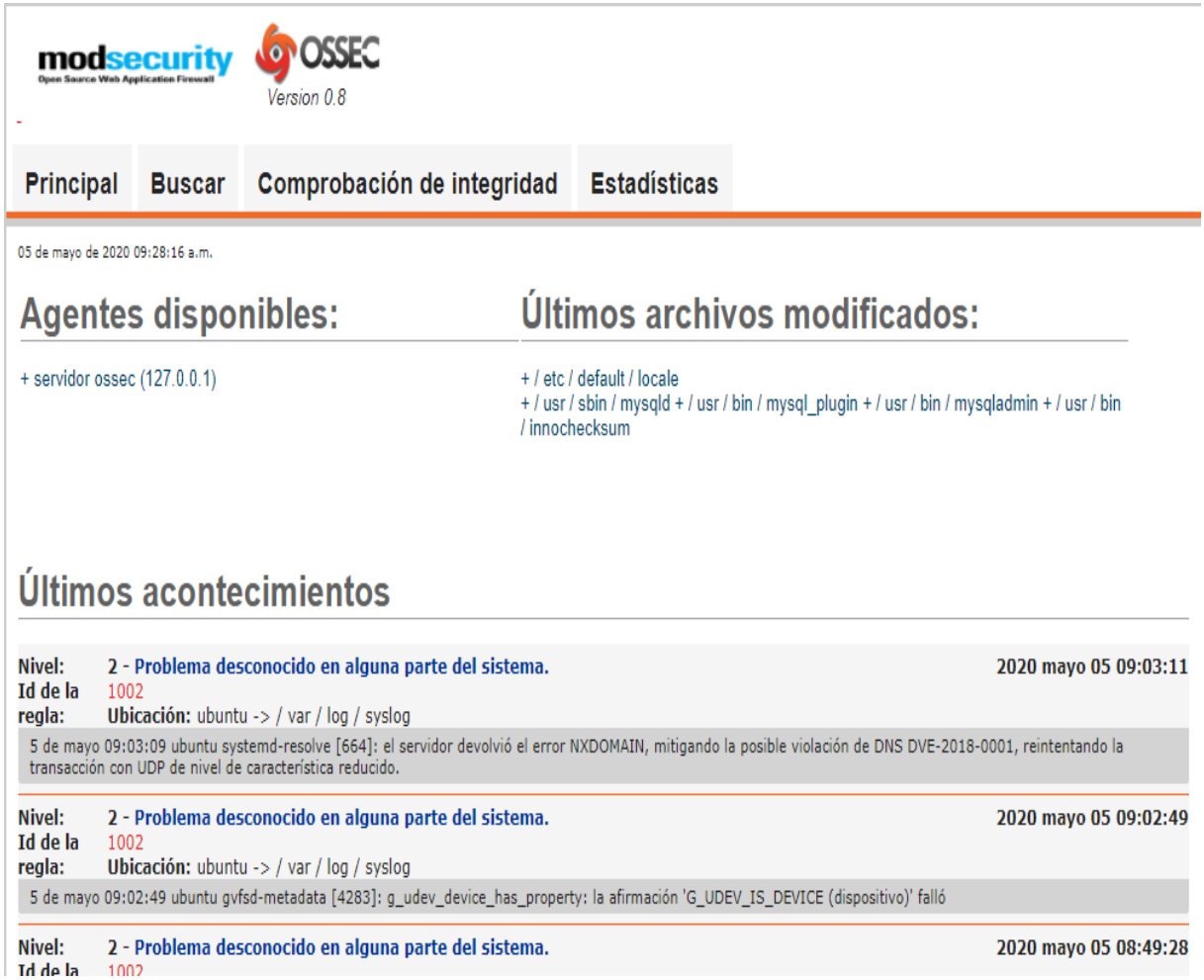


Figura 3-10.: Prototipo de Framework MODSEC, **Fuente:** Elaboración Propia

3.3. Evaluación del Prototipo de Framework Propuesto.

En la Figura **3-11** se muestra el ciclo PHVA.

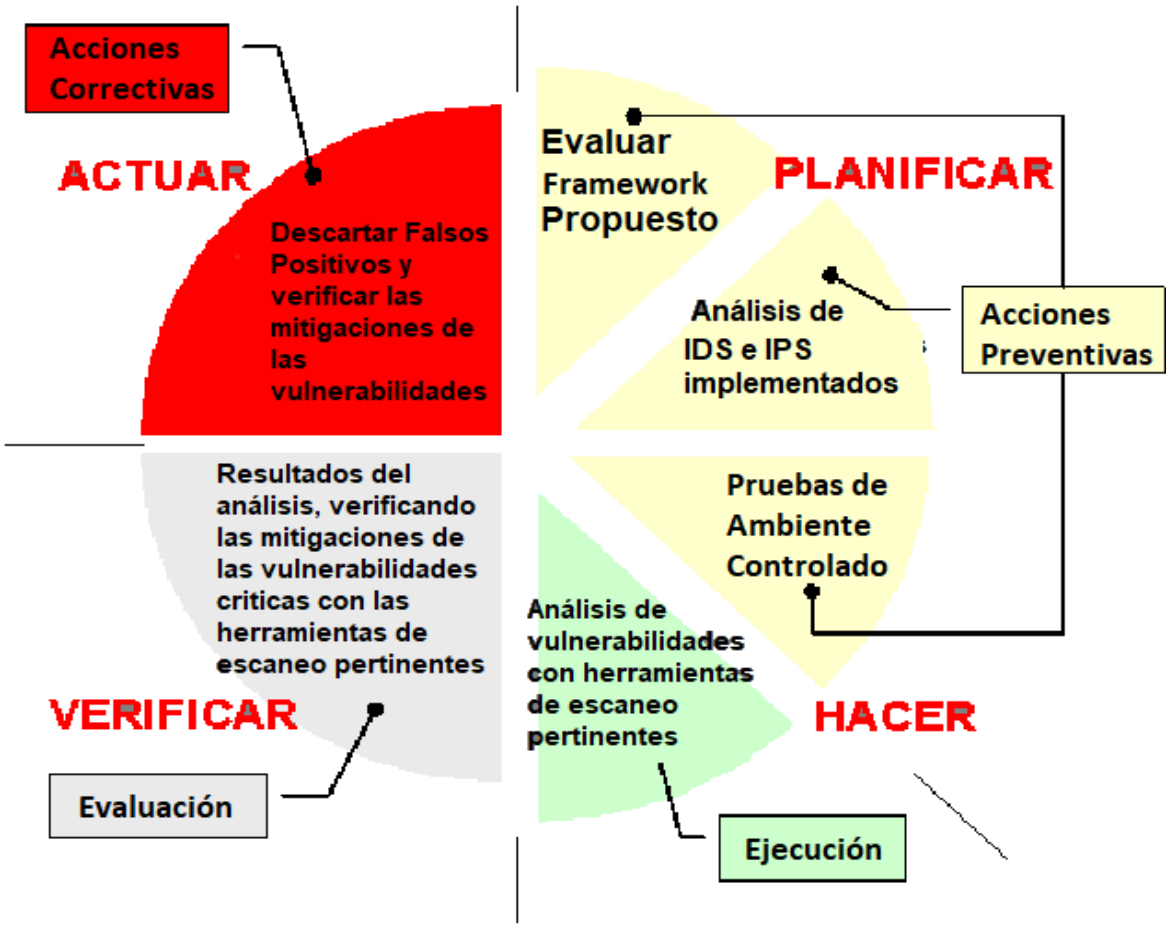


Figura 3-11.: Ciclo Desarrollo de Evaluación del Framework, **Fuente:** Elaboración Propia

En esta fase se realizan nuevas pruebas bajo ambiente controlado con la herramienta wireshark para el escaneo de la red y así poder esniffar los paquetes encontrados y ver si los paquetes ahora viajan de manera segura por la red de una entidad hospitalaria sistema 1 hacia la segunda entidad hospitalaria Sistema 2, dichos paquetes deben viajar de manera segura para proveer confidencialidad en sus datos a los pacientes y así llegar a obtener mejores prácticas de seguridad en las entidades hospitalarias. Se utiliza el motor MIRTH CONNECT nuevamente para el envío de archivos HL7 CDA R2 con las mismas configuraciones del primer Análisis que se realizó en la FASE UNO, ver Anexo B.
Se quiso con este último análisis comprobar si la implementación del prototipo de FRAMEWORK funciona para la mitigación de las vulnerabilidades en la confidencialidad del

estándar HL7 CDA R2. A esto se sumó un nuevo ataque por medio de análisis de vulnerabilidades con la herramienta ACUNETIX en su versión Gratuita a la plataforma del prototipo de framework activado y así poder verificar si la implementación de este ha sido efectivo o no, el informe se puede descargar del Anexo F, para mitigar las vulnerabilidades del estándar HL7 CDA R2 se activan los módulos y se activa la plataforma MODSEC con apache para que quede en funcionamiento en tiempo real, ver Anexo G

4. RESULTADOS

4.1. PRIMERA FASE, Identificar Problemas y Determinar Herramientas.

En este primer acercamiento para conocer el trabajo con el estándar HL7 se contactó con un profesional en el área al cual se le aplicó una entrevista para conocer el trabajo que se ha venido haciendo hasta el momento en Colombia, precisamente en la ciudad de Medellín, esto da una visión real y certera de las complicaciones de seguridad que este estándar ha tenido y así conocer un poco más a fondo el comportamiento del estándar. Cuando se aplicó la entrevista al profesional en el estándar, aclaro que en la promotora médica Las Américas de la ciudad de Medellín solo se trabaja con el estándar HL7 en su versión 2.0 y se desconoce del estándar HL7 CDA R2, simplemente utilizan HL7 para envío de mensajes como datos personales, pero no estructurado como un historial médico electrónico. En Colombia hasta el momento no se ha trabajado con el estándar HL7 CDA R2 por desconocimiento de él y por falta de profesionales conocedores en el tema. Los resultados de la entrevista que se encuentra en el Anexo H dan a conocer que a pesar de tomar medidas de seguridad básicas en la capa de aplicación aún se conservan Incidentes a tratar.
Se consultaron los principales Incidentes de HL7 y CDA R2 tomando como referencia los análisis realizados por la empresa SMART mencionados en la metodología:

- Ejecución remota de código.
- Comunicaciones de Texto Claro.
- Carece de autenticación.
- interceptación de datos.
- alteración de los datos.
- Pérdida de Autenticación y Gestión de Sesiones.
- fuga de información

Para poder Identificarlos se realizó una matriz en formato Check list para correlacionar los principales puntos de afectación de estos Incidentes de seguridad del estándar Hl7 CDA R2 para posteriormente caracterizarlos en ataques, ver Tabla **4-1**.

PROBLEMAS	CONFIDENCIALIDAD	OWASP	HL7 CDA R2
Ejecución remota de código	X	X	X
Comunicaciones de Texto Claro	X		
interceptación de datos	X	X	X
Pérdida de Autenticación y Gestión de Sesiones.X			X
fuga de información	X	X	X
alteración de los datos.	X	X	X

Tabla 4-1.: Identificación de los Incidentes bajo principales puntos de afectación en HL7 CDA R2, **Fuente:** Elaboración Propia

De la Tabla **4-1** se escogieron 4 Incidentes, estos 4 fueron los que cumplieron con las 3 características completas que afectan la confidencialidad, que se encuentren en el top 10 de OWASP y que sean incidentes que afecten el estándar HL7 CDA R2 los cuales son:

- Ejecución remota de código.
- interceptación de datos.
- alteración de los datos.
- fuga de información

Los problemas para esta investigación solo se centran en parte de la problemática como las comunicaciones en texto claro, cuando es enviado el mensaje HL7 en un tráfico de wireshark se puede ver el mensaje sin cifrar como lo muestra la Figura **4-1**.

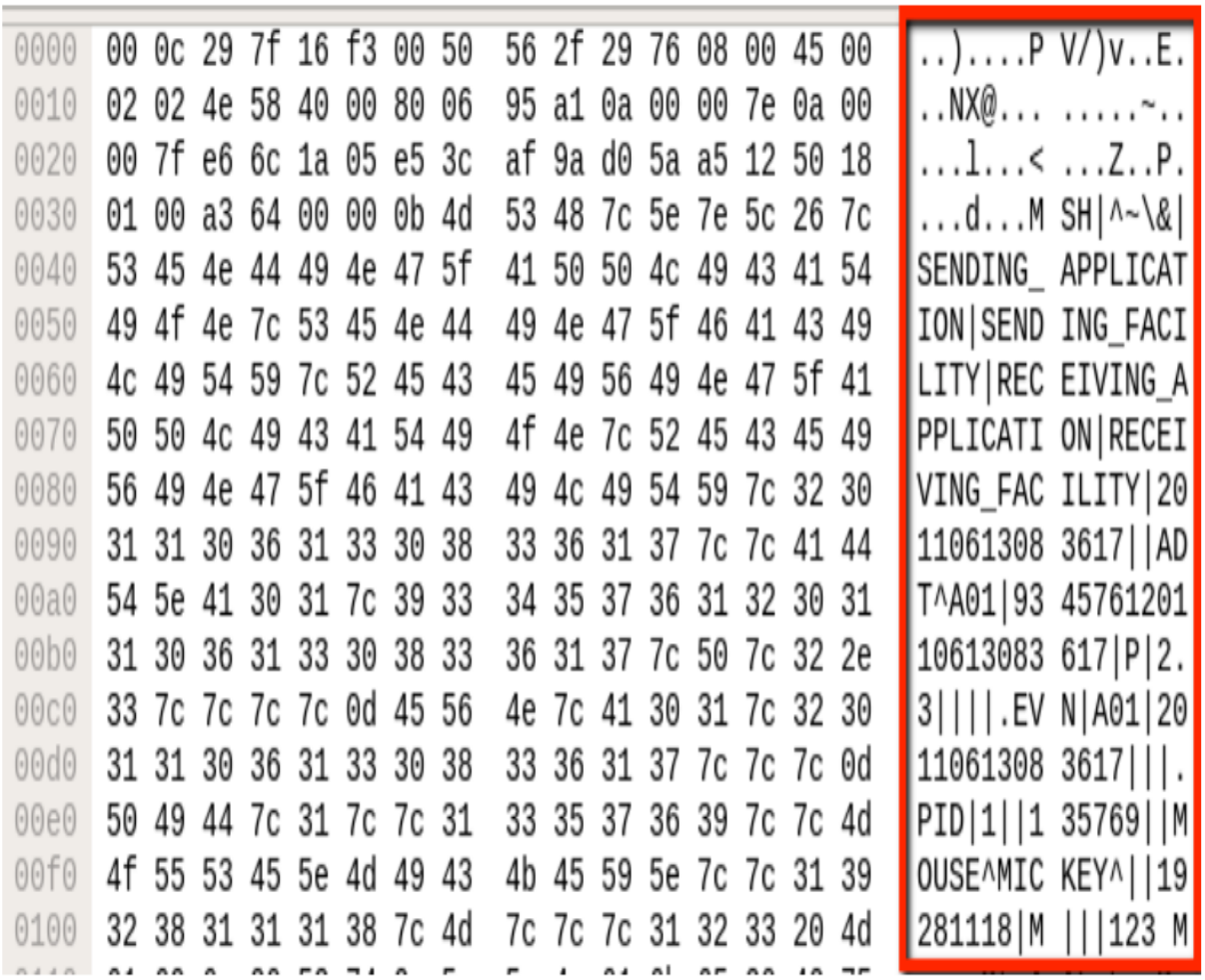


Figura 4-1.: Datos HL7 sin cifrar recopilados de Wireshark en un sistema MITM atacante, **Fuente:** SANS Institute Information Security Reading Room [28]

Este problema existe para el estándar HL7 CDA R2 ya que el historial médico está estructurado en un archivo XML el cual sigue siendo plantilla de datos claros y en wireshark se puede mostrar el contenido en texto claro.

Después de Identificar los posibles Incidentes del estándar HL7 CDA R2 se caracterizan y se clasifican en ataques como se muestra en la Figura **4-2**.

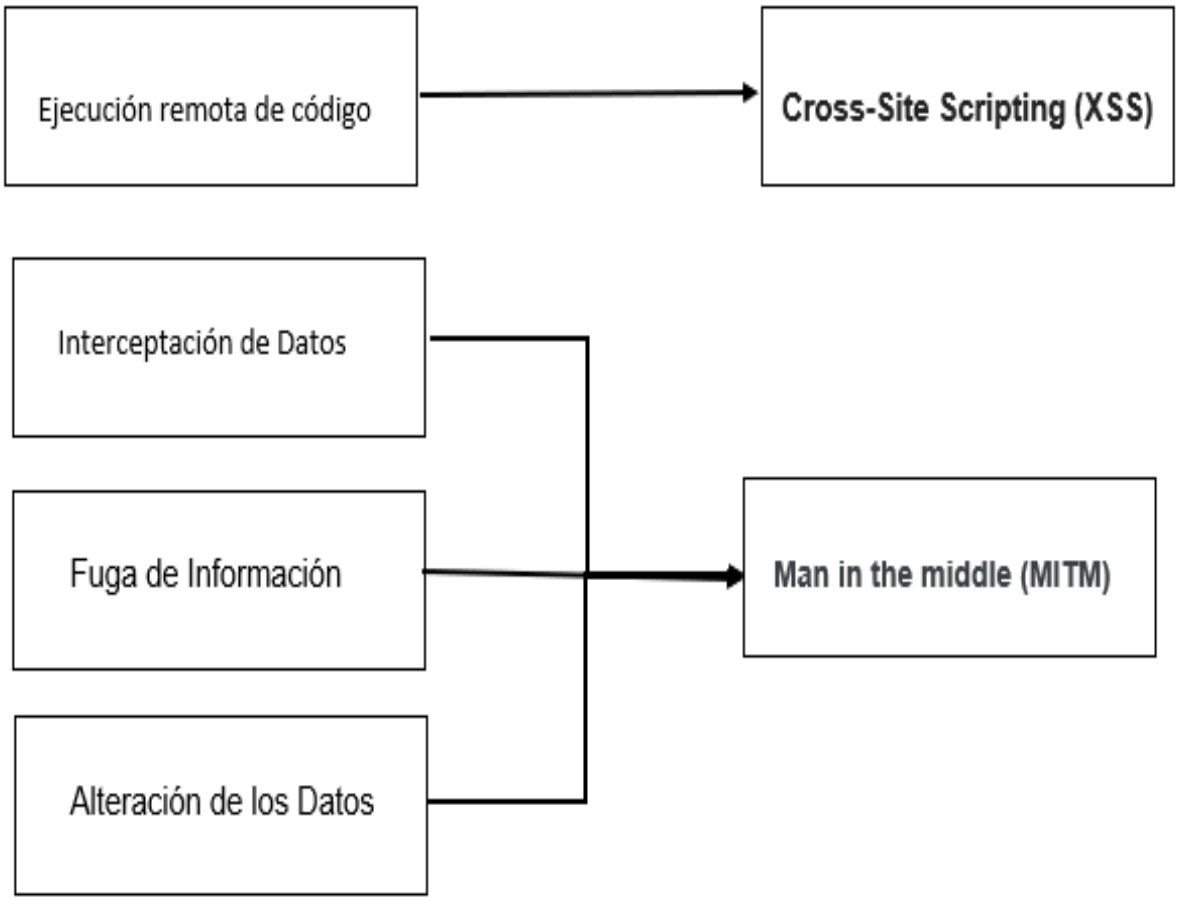


Figura 4-2.: Clasificación de incidentes en ataques, **Fuente:** Elaboración Propia

En la Figura **4-2** se clasifican los incidentes con su respectivo ataque, el XSS es una ejecución remota de código, donde se puede redireccionar archivos a paginas externas o implementar mensajes de alerta de criminales, la interceptación de datos, fuga de información y alteración de estos datos son posibles por el ataque de Hombre en el medio o MITM (Man in the Middle), este ataque que consiste en interceptar la información antes de que llegue a su destinatario, de esta forma se cumplen los dos primeros incidentes interceptación y fuga de datos y lógicamente después de tener estos datos se podría alterar la información y redirigirla a su destino pero con los datos cambiados, estos ataques ya los tiene reportados la **CVE**. Al encontrar ya los ataques efectivos se procedió a buscar el motor de envío de mensajería en HL7, para ello se realiza una tabla comparativa entre los mejores motores, basado en Características claves para así escoger la más idónea en esta investigación como lo muestra la Tabla **4-2**.

Motores de Envío mensajería HL7	INTEROPERABILIDAD	Open Source	Fácil Implementación	Múltiples Formatos de Mensajería	Compatible con CDA	Compatible con XML	Seguridad básica
Iguana	x			x	x	x	x
DICOM PACS	x				x	x	x
E-Resetar	x		x				x
HL7 Soup	x			x			
quick view hl7	x	x	x				
Smart HL7 Tools Bundle	x			x	x	x	
Mirth Connect	x	x	x	x	x	x	x

Tabla 4-2.: Motores de envío de mensajería HL7, **Fuente:** Elaboración Propia

Las pautas de verificación para escoger este motor de envíos de mensajes HL7 fueron esenciales ya que estas definieron cual fue el mejor motor. Para esta investigación se tomó las características principales que sea Open Source, la compatibilidad con CDA y XML, seguridad básica y los múltiples formatos de Mensajería, es así como después de probar todos estos motores el escogido fue **MIRTH CONNECT**, se escoge este motor ya que cumple con todas las características expuestas en la Tabla **4-2** y porque tiene unas excelentes ventajas para la implementación.
las ventajas de este motor Mirth Connect son:

- Licencia gratuita.
- Fácil implementación.
- Creación de canales sencilla.
- Emplea varios protocolos de transporte.
- Emplea múltiples formatos de mensajería.
- Reduce significativamente los tiempos de desarrollo e implementación de interfaces.
- Reduce significativamente los problemas asociados al mantenimiento de interfaces.

Este motor es considerado la navaja suiza en envíos de mensajería de HL7 ya que soporta múltiples plataformas, varios motores de Bases de Datos y Protocolos, metodologías de transporte y formatos de presentación de datos soportados.
Se procede a realizar el modelo de red HL7 CDA punto a punto para esta investigación, en la Figura **4-3** se puede apreciar este modelo simple por medio de canal de distribución

creado por el motor de envíos MIRTH CONNECT por donde se enviará el historial médico electrónico desde Sistema 1 hacia Sistema 2 vía WIFI, así se obtendrá una lectura más acercada a la realidad.

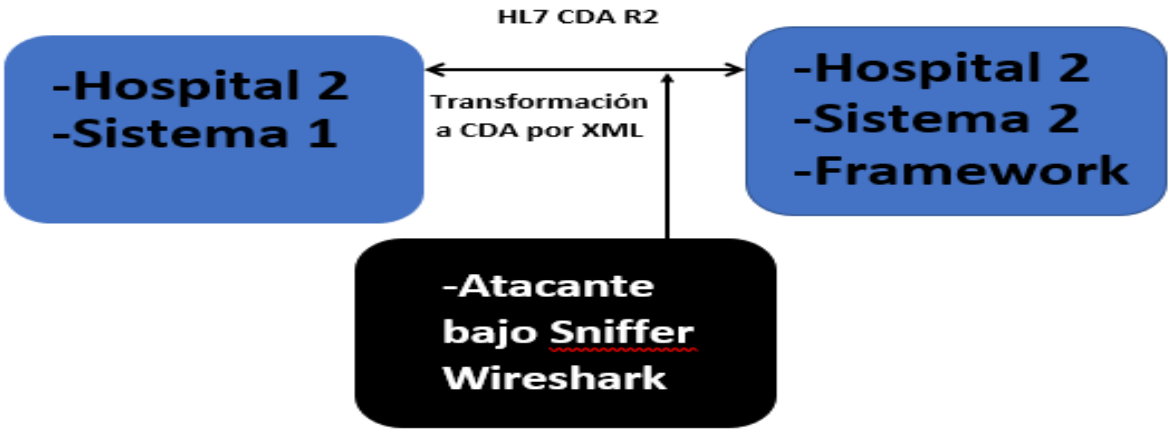


Figura 4-3.: Modelo de red HL7 CDA R2 Punto a Punto, **Fuente:** Elaboración Propia

Se instaló el motor de envíos Mirth Connect Anexo A para proceder a la creación del canal de envío de los historiales médicos, el canal (Entidad Hospitalaria 1) se llama Sistema 1 y el destino (Entidad Hospitalaria 2) se llama Sistema 2. Se envía un historial médico en formato XML por el canal hacia la entidad 2 donde se podrá visualizar este Historial, para la creación de mensajes en HL7 existen varias herramientas como:

- HL7 SOUP Editor.
- 7 Spy.
- Interface Explorer.
- 7edit.
- 7Scan.
- XML Spy.
- Quick view HL7.
- Smart HL7.

Para entidades mucho más grandes como hospitales de ciudades principales en Europa se implementó una red de HL7 CDA R2 con el motor de envíos e interfaz con una estructura como se muestra en la Figura **4-4**, esa red muestra las principales dependencias por donde viajaran los historiales médicos bajo un motor de interfaz y envíos MIRTH CONNECT.

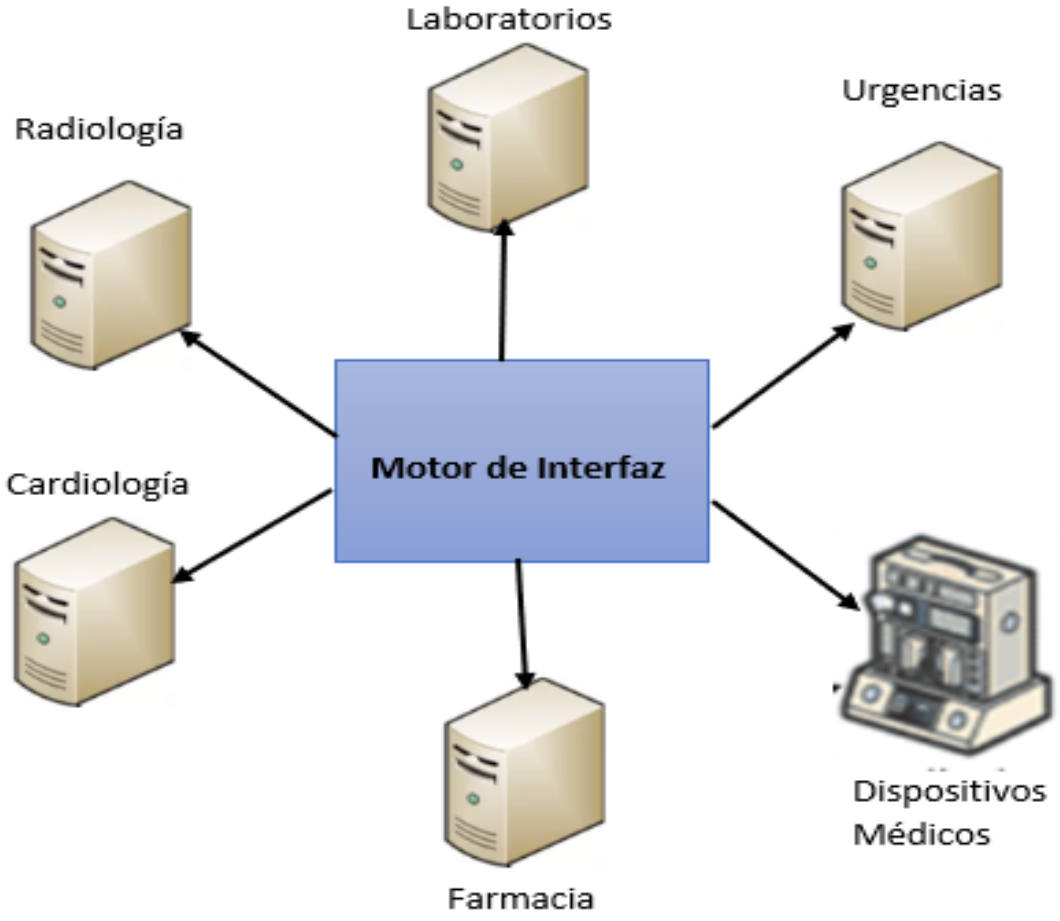


Figura 4-4.: [Modelo de red HL7 CDA R2 Motor de Interfaz, **Fuente:** Elaboración Propia

Cuando se utiliza un editor de HL7 este nos creara una estructura de mensaje en HL7 V2 ó HL7 V3 como se miró en el marco teórico en la Figura **2-1**, esta estructura es diferente al de historiales médicos el cual utiliza la plantilla de estructura xml que la provee la empresa HL7 de forma gratuita llamada CDA.xsl como se muestra en la Figura **4-5**. En el Anexo C se encuentra el link de descarga para ver la estructura completa del archivo CDA.xsl.

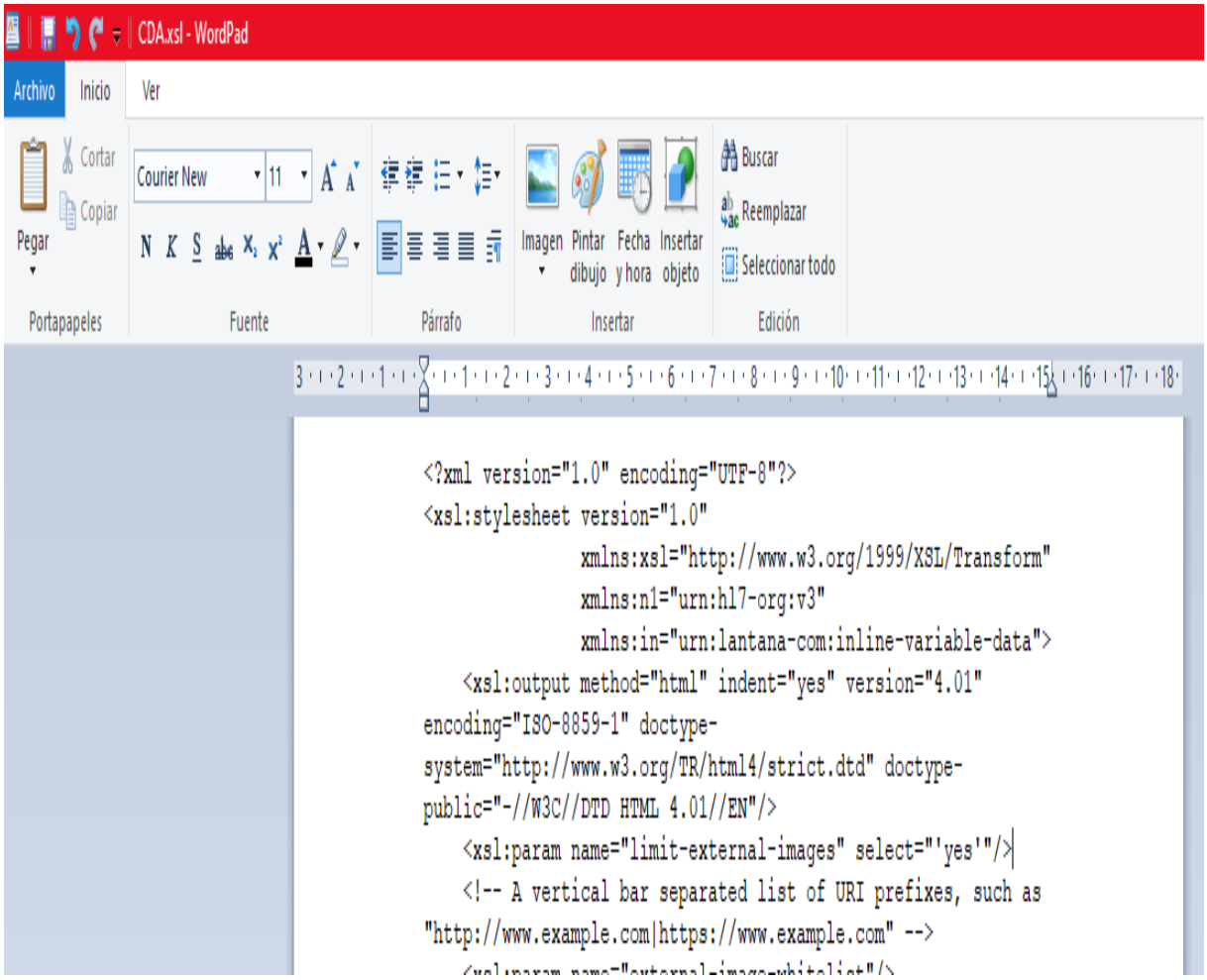


Figura 4-5.: Estructura de Archivo CDA.xsl, **Fuente:** Elaboración Propia

La estructura de la plantilla CDA se encuentra en XML y se divide en 2 partes, la cabecera (header) y el cuerpo (body) como la Figura **4-6** lo muestra, esta cabecera tiene información de meta-datos sobre el documento e información como la fecha, la identificación, el título, el idioma, también tiene información como los participantes del documento como el autor, el paciente y l proveedor.

```
<ClinicalDocument xmlns="urn:hl7-org:v3" classCode="DOCCLIN" moodCode="EVN">
    <!-- ************************************************************
              ENCABEZADO DEL DOCUMENTO ELECTRÓNICO - HL7 CDA R2
         ************************************************************ -->
    <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
    <!-- Atributos del encabezado -->
    <templateId/>
    <id/>
    <code/>
    <title/>
    <effectiveTime/>
    <confidentialityCode/>
    <languageCode/>
    <setId/>
    <versionNumber/>
    <copyTime/>
    ....
    <component typeCode="COMP" contextConductionInd="true">
        <!-- ************************************************************
                  CUERPO DEL DOCUMENTO ELECTRÓNICO - HL7 CDA R2
             ************************************************************ -->
        <structuredBody>
            <component>
                <section>
                    <entry></entry>
                    <entry></entry>
                    <entry></entry>
                </section>
            </component>
            <component></component>
            <component></component>
        </structuredBody>
    </component>
</ClinicalDocument>
```

Figura 4-6.: Estructura de CDA en 2 partes, **Fuente:** Elaboración [39]

El cuerpo ya contiene la información del documento, la información siempre debe contener una parte textual que asegure la legibilidad humana. Este cuerpo está dividido por niveles:

- NIVEL 1. Esta tiene un contenido no estructurado, por ejemplo, puede contener un documento pdf incrustado.
- NIVEL 2. Este nivel ya tiene un contenido estructurado y codificado en secciones.
- NIVEL 3. En este nivel el contenido ya es completamente codificado y estructurado.

Estas partes de la estructura del documento CDA se pueden apreciar mucho mejor en la Figura **4-7**.

Figura 4-7.: Estructura de CDA en 2 partes, **Fuente:** Elaboración [12]

4.1.1. Determinar Herramientas idóneas de detección y prevención de vulnerabilidades.

Después de saber cuales son los Incidentes a tratar y saber que son vulnerabilidades del top ten de OWASP se realizó una investigación de herramientas por medio de artículos y de experiencia propia para detectar estos Incidentes y tomar acciones para mitigarlos, se encontró una gama de herramientas con diferentes opciones así como de paga y open source, para este proyecto en particular se optó por la característica principal que sea OPEN SOURCE para tener fácil acceso y de bajo consumo de recursos, en la Tabla **4-3** se puede apreciar las características, las herramientas encontradas y cuales fueron las 2 herramientas seleccionadas.

HERRAMIENTAS	Open Source	Descarga Fácil	Fácil Implementación	Capa 7
Fortinet (IPS)			x	x
Mod Security (IPS)	x	x	x	x
Nozomi SCADAGuardian (IDS)			x	
Cisco NGIPS (IPS)			x	x
Corelight y Zeek (IDS)		x		x
OSSEC (IDS)	x	x	x	x
Fidelis Network (IPS)		x		x
FireEye Intrusion Prevention System (IDS)-(IPS)		x		
SNORT (IDS)	x	x		

Tabla 4-3.: Tabla de los mejores IDS e IPS, **Fuente:** Elaboración Propia

Como anteriormente se ha mencionado, la principal característica que se tenía en cuenta es que sea OPEN SOURCE, acompañado de que trabaje para la capa 7 y que sea de fácil implementación, dado por entendido que la fácil implementación se basa en lo relativo del conocimiento y entendimiento para poder instalar estas herramientas, para esta investigación ya se tenían conocimientos previos de los dos módulos escogidos. Se obtuvo el resultado de 3 Herramientas Virtuales que podrían trabajar, pero se descartó SNORT por complicada implementación y que trabaja sobre la capa de red en tiempo real pero no maneja plataforma de usuario donde correlacionar los ataques en los logs y mostrarlos, dando así por Resultado Dos Herramientas: MOD SECURITY (IPS), OSSEC (IDS). Estas herramientas se trabajaron en el transcurso de la finalización de semestre de esta maestría, es así como surge una pregunta investigativa para desarrollar en este proyecto, las 2 herramientas podrían trabajar en conjunto dejando así un sistema bastante seguro de implementación OPEN SOURCE de trabajo IDS e IPS. Es cuando se puede verificar que las dos herramientas trabajan muy bien en conjunto teniendo una buena instalación y configuración, es por eso que se decide unificar estas herramientas para crear así un prototipo de Framework que pueda mitigar las vulnerabilidades existentes.

4.2. SEGUNDA FASE, Desarrollo del Prototipo de Framework

OBTENCIÓN DE HISTORIALES MÉDICOS ELECTRÓNICOS HL7 CDA R2

Para la implementación de este framework se hizo primeramente, la obtención de los Historiales médicos en formato de plantilla XML para el estándar HL7 CDA R2 de muestra para poder realizar los envíos para el análisis, esto se realizó por una conexión directa con la empresa HL7 de Estados Unidos, la empresa optó por brindar algunos historiales médicos y la plantilla CDA.xsl la cual hace que el documento médico sea convertido en Historial médico electrónico, esto se realizó en conjunto con el ganador del concurso Software Medical Viewer realizado en enero del 2017 por la empresa HL7 para desarrollar un visor de este tipo de documentos HL7 CDA R2 llamado Backbeach Software Medical Record Viewer como se puede ver en la Figura **4-8**.

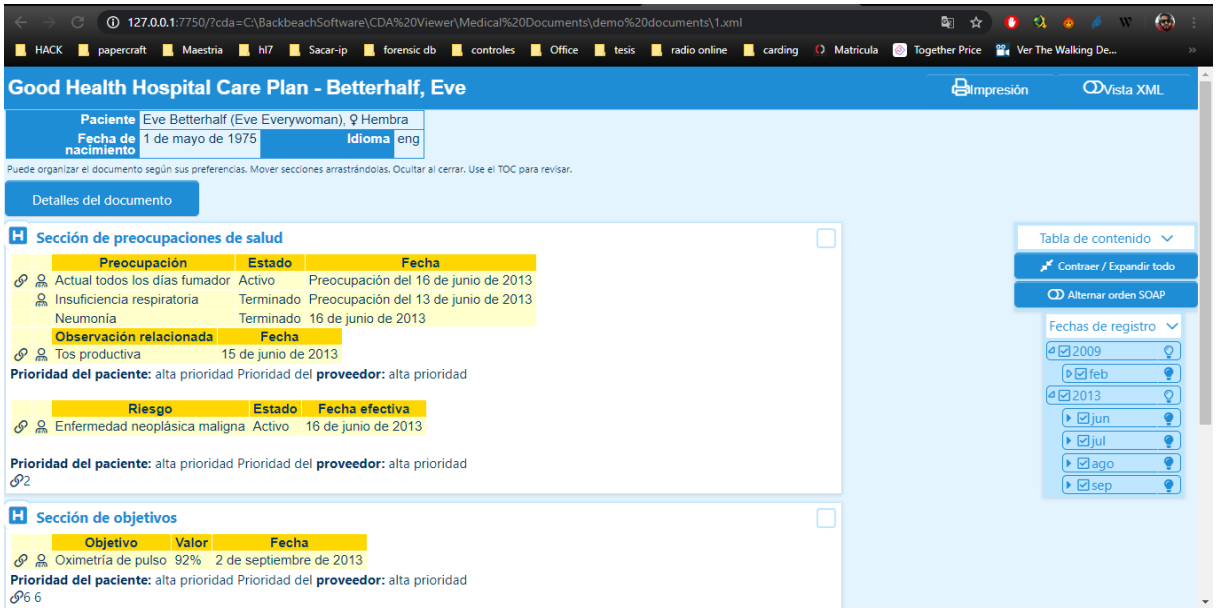


Figura 4-8.: Backbeach Software Medical Record Viewer, **Fuente:** Elaboración backbeach software[38]

INSTALACIÓN DEL MOTOR MIRTH CONNECT Y ENVÍO DE MENSAJES HL7 CDA R2

El siguiente paso a seguir, después de la instalación del motor de envíos MIRTH CONNECT Anexo A, fue la creación del canal de transmisión Anexo B, para realizar el envío de estos Historiales Médicos Electrónicos HL7 CDA R2. El envío se realiza de una manera simple para comprobar que el archivo si está viajando a la carpeta destino. Se tienen los Historiales CDA guardados en una carpeta llamada para enviar como se puede apreciar en la Figura **4-9**, Se copia el archivo 1.xml para ser pegado en la carpeta donde se realizará el envío llamada ejemplo como se ve en la Figura **4-10**, el archivo 1.xml al ser copiado en la carpeta ejemplo este inmediatamente por las configuraciones que se realizaron en el canal de envío después de 5 segundos desaparece Figura **4-11** y se transmite a la carpeta contenedora DESTINO la cual se llama demo documents Figura **4-12**, que es la carpeta que está configurada por el programa Visor Backbeach Software Medical Record Viewer para recibir estos archivos y poder así observarlos de manera óptima en la WEB.

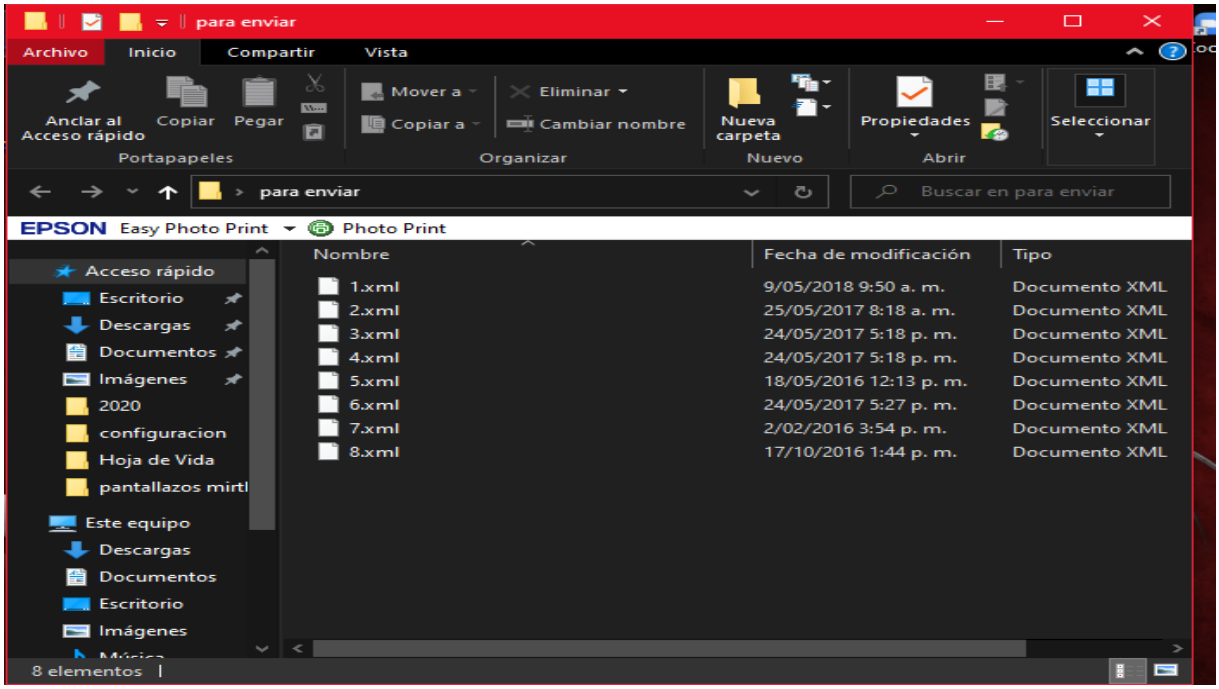


Figura 4-9.: Carpeta de historiales Médicos CDA, **Fuente:** Elaboración propia

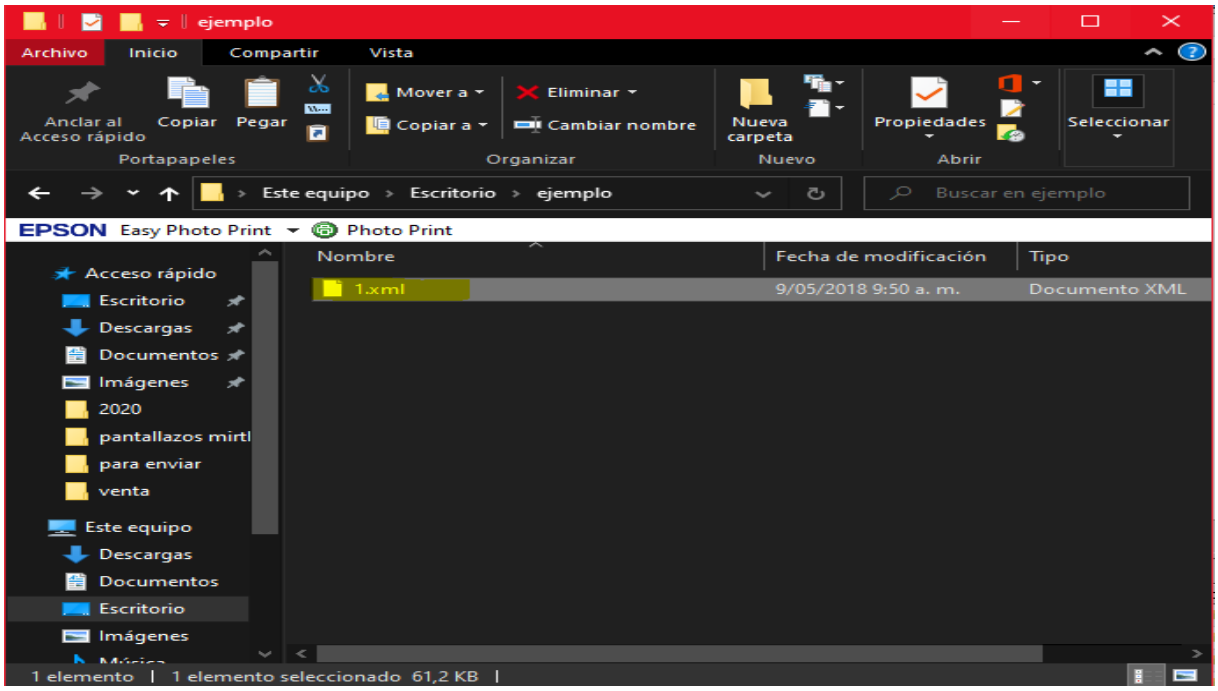


Figura 4-10.: Carpeta donde se realizará el envío, **Fuente:** Elaboración propia

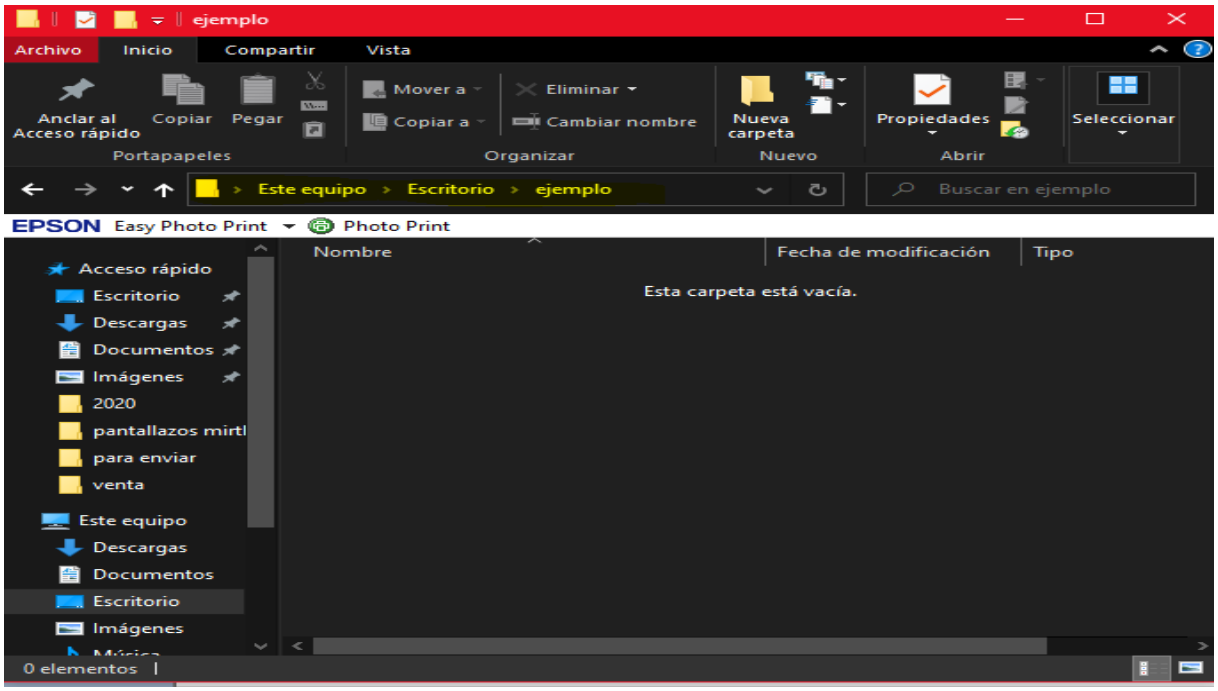


Figura 4-11.: Desaparece el archivo CDA enviado, **Fuente:** Elaboración propia

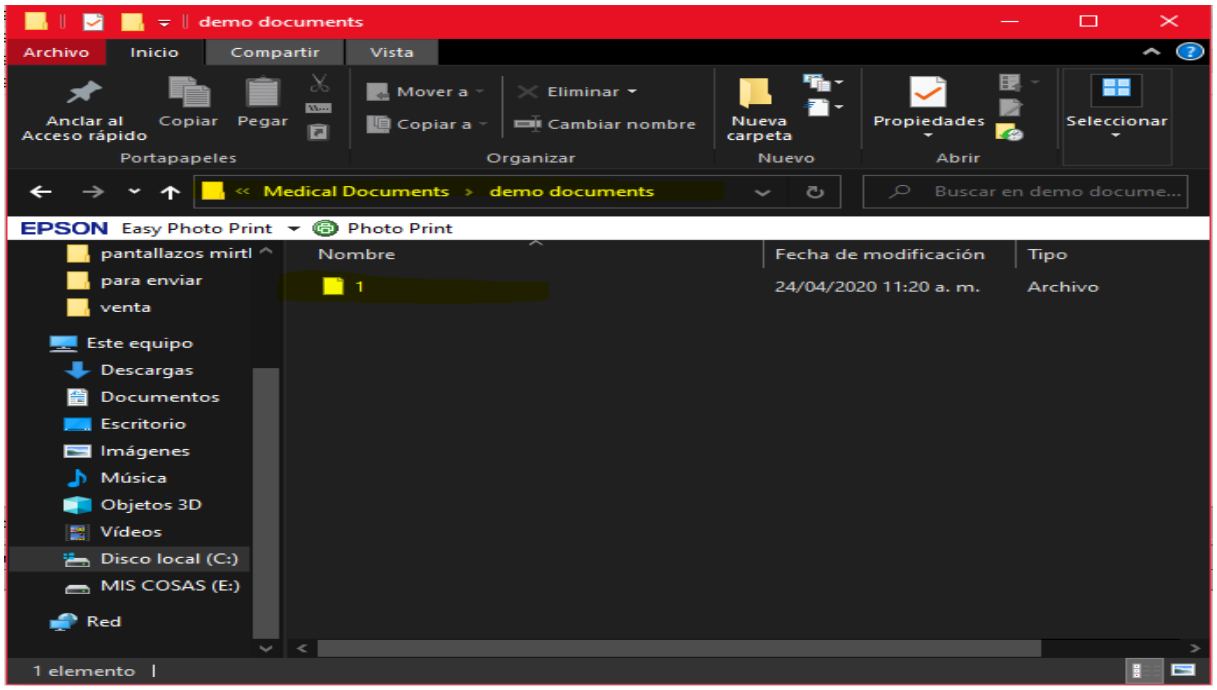


Figura 4-12.: Carpeta Receptora DESTINO de los Archivos HL7 CDA R2, **Fuente:** Elaboración propia

Al ver que el archivo desaparece de la carpeta ejemplo y aparece en la carpeta demo documents se podría asegurar que el canal de MIRTH CONNECT funciona de manera óptima, por lo tanto, se debe revisar el DASHBOARD del motor de envíos para así poder corroborar que el envío se realizó de manera eficaz y exitosa Figura **4-13**, de esta manera ya se puede ASEGURAR que el envió fue totalmente exitoso, se entra al panel de confirmación donde se observa los datos que fueron enviados Figura **4-14**, y por último se observa el mensaje exitoso que el archivo viajo y fue recibido de manera efectiva Figura **4-15**.

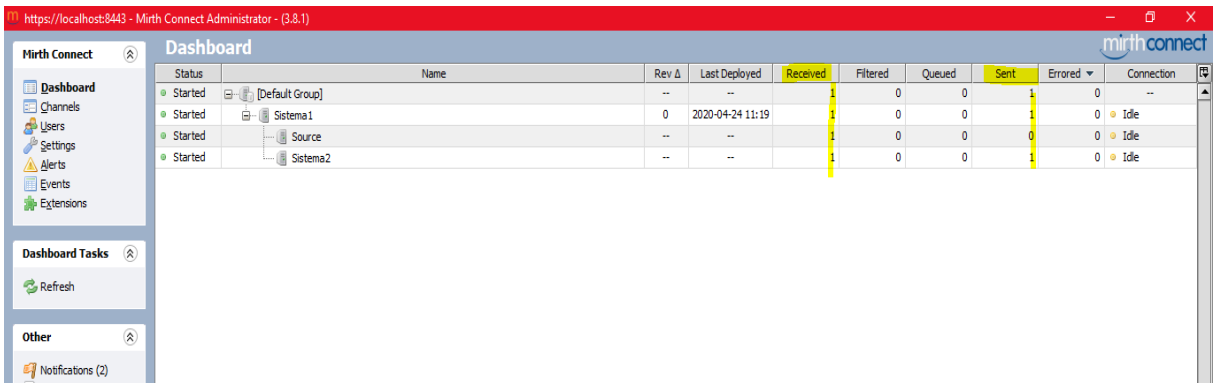


Figura 4-13.: Confirmación en MIRTH CONNECT de envío EXITOSO, **Fuente:** Elaboración propia

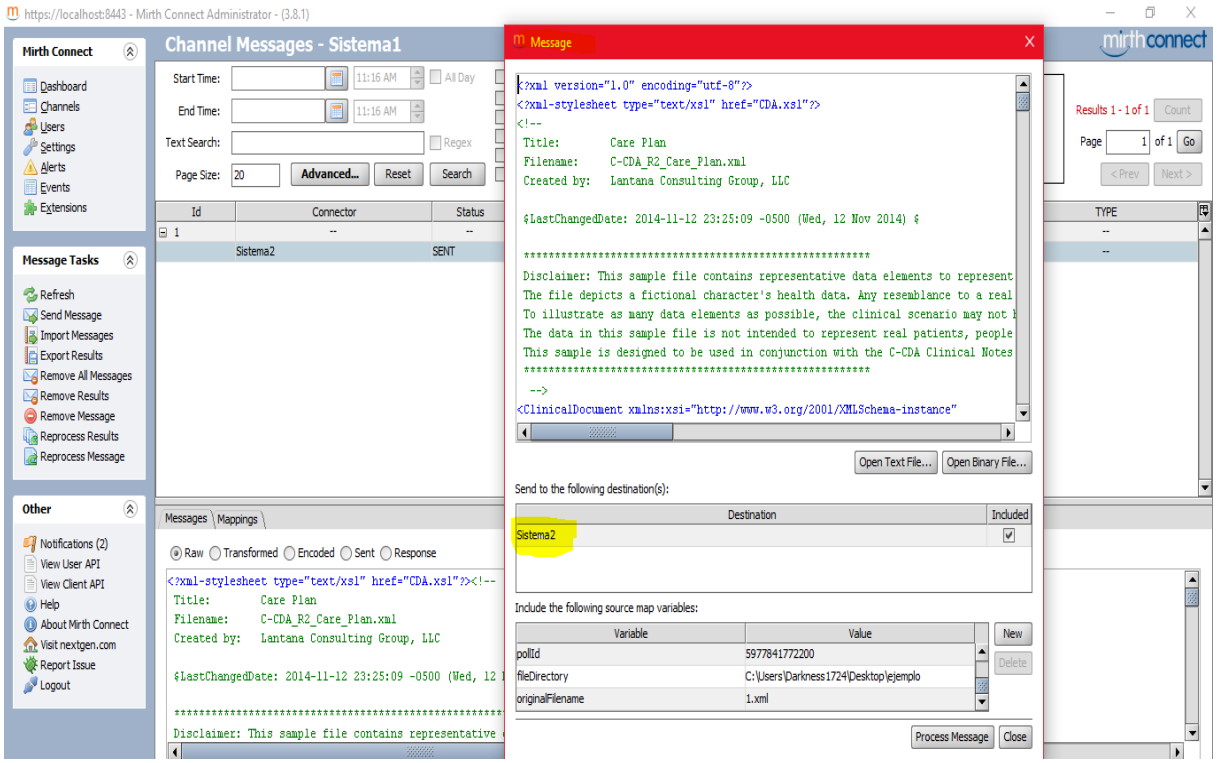


Figura 4-14.: Contenido del Mensaje que se envió, **Fuente:** Elaboración propia

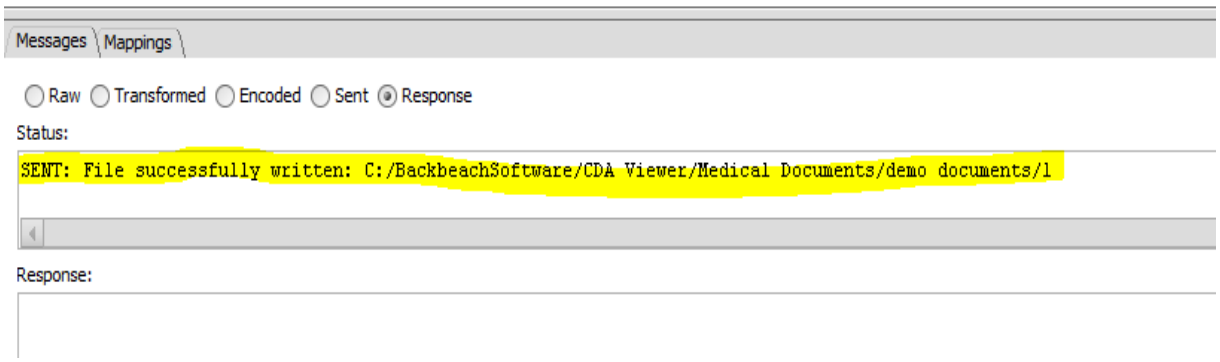


Figura 4-15.: Mensaje de envío exitoso a carpeta DESTINO DEMO DOCUMENTS, **Fuente:** Elaboración propia

Al completar el envío se realizó un cambio en el motor MIRTH CONNECT en la parte de DESTINO en el canal de envíos para que el documento viaje por conexión HTTP, solo se agregaron unos parámetros de mensajería para que el archivo viaje con el nombre original y la extinción .XML, como se puede observar en la Figura **4-16**, bajo el método POST, la dirección IP del equipo destino es 192.168.1.67 que viajará por el puerto 666 como lo muestra la Figura **4-17**, y . Es aquí donde se encuentra el fallo de seguridad.

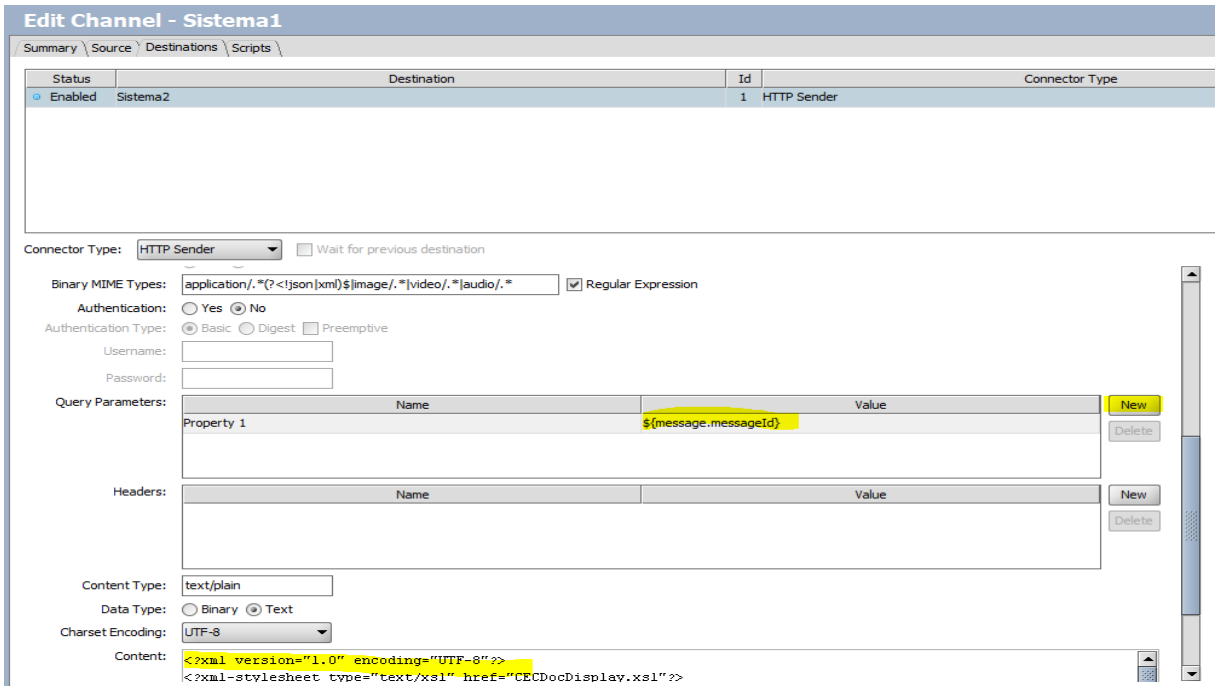


Figura 4-16.: Cambios de método de conexión al enviar los CDA, **Fuente:** Elaboración propia

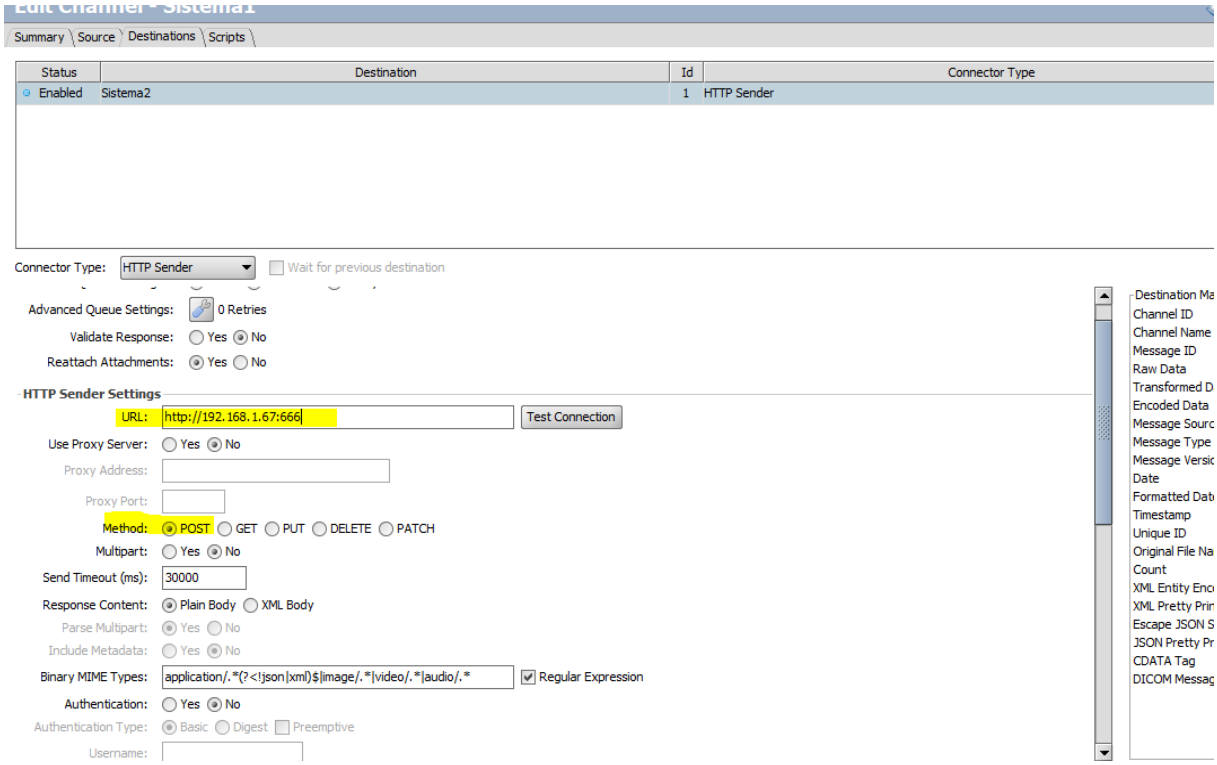


Figura 4-17.: Envío DESTINO por método POST en HTTP SENDER, **Fuente:** Elaboración propia

ATAQUE DE MAN IN THE MIDDLE(Hombre en el Medio o MitM)

El ataque MitM bajo ambiente controlado empieza al realizar el envío de los historiales médicos electrónicos HL7 CDA R2 hacia la entidad destino SISTEMA 2 por conexión HTTP SENDER en método POST. Se realizó un escaneo de red por medio de una técnica hacking conocida como sniffer la cual conociste en escuchar todo lo que pasa por una red bajo herramientas elaboradas para este fin y poder aprovecharse de vulnerabilidades encontradas, la herramienta utilizada en esta investigación fue WIRESHARK [40]. Los parámetros del ataque a tener en cuenta fueron:

- IP DESTINO y PUERTO 192.168.1.67:666.
- IP DE ATACANTE 192.168.1.51.
- Herramienta de SNIFFER WIRESHARK.
- Motor de Envíos MIRTH CONNECT.

Se puede observar en la Figura **4-18** la ip Atacante y en la Figura **4-19** la ip destino, se procedió a realizar el envío de los archivos CDA nuevamente por el motor MIRTH CONNECT

y el envío se realizó de forma oportuna como se muestra en la Figura **4-20**, en la herramienta WIRESHARK se empezó el análisis de la red WIFI con todas sus navegaciones y varios equipos conectados para hacer el ambiente controlado lo más real posible. Después del envío la herramienta wireshark realizó su trabajo y después de 3 minutos de captura de paquetes esta herramienta se puso en STOP y en la parte de filtros de wireshark Figura **4-21** se escribe el filtro ip.dst==192.168.1.67 y tcp.port == 666 el cual va a filtrarnos todos los paquetes que están viajando hacia esa ip destino, después de un análisis de los paquetes se encontraron los que fueron enviados con el contenido de los archivos CDA como se muestra en la figura Figura **4-22**, se puede apreciar varios paquetes segmentados con todo el contenido del archivo xml HL7 CDA R2.

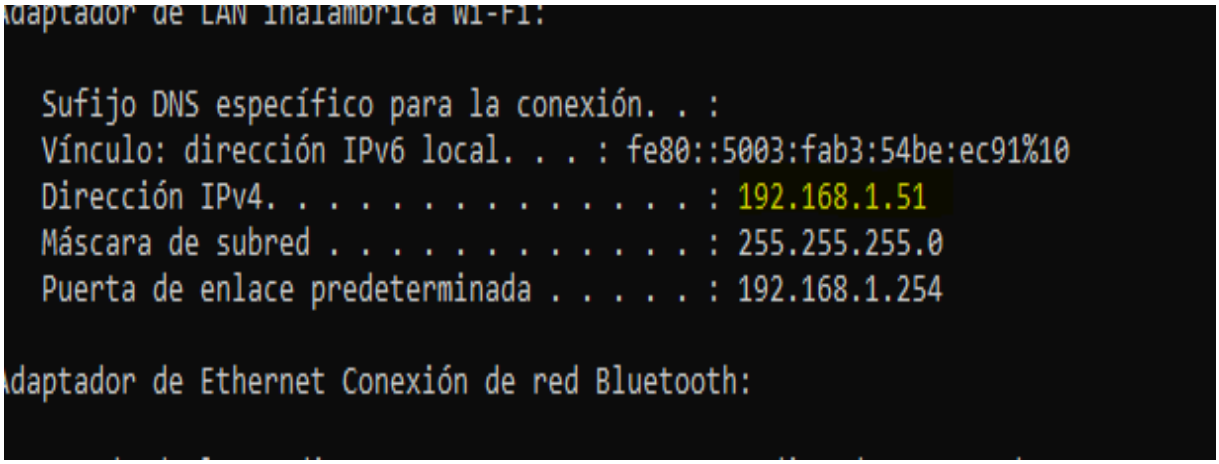


Figura 4-18.: IP ATACANTE bajo ambiente controlado, **Fuente:** Elaboración propia

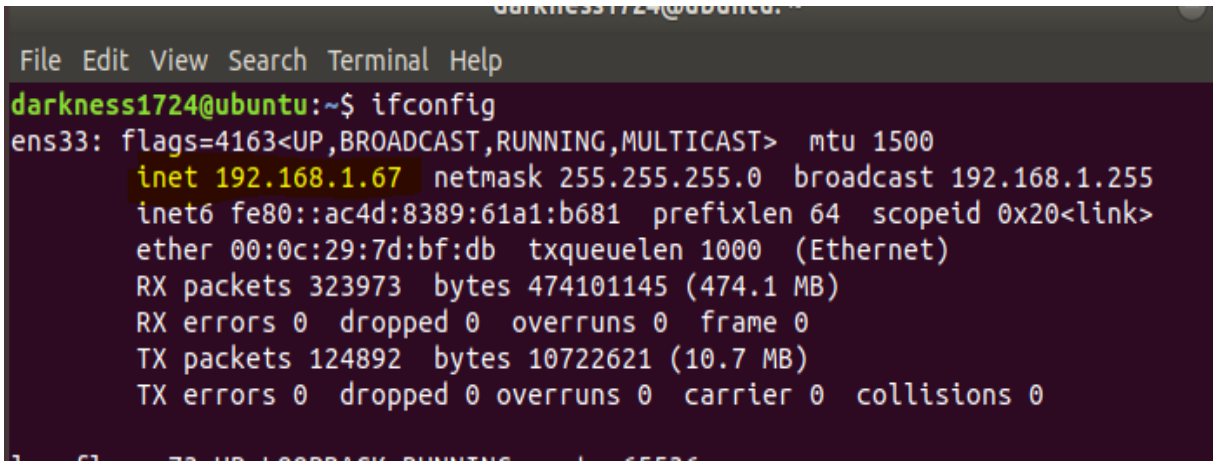


Figura 4-19.: IP DESTINO bajo ambiente controlado, **Fuente:** Elaboración propia

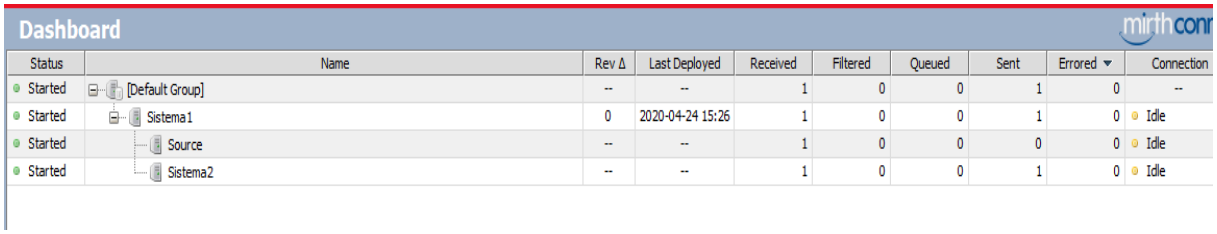


Figura 4-20.: Envío Exitoso por método HTTP SENDER, **Fuente:** Elaboración propia

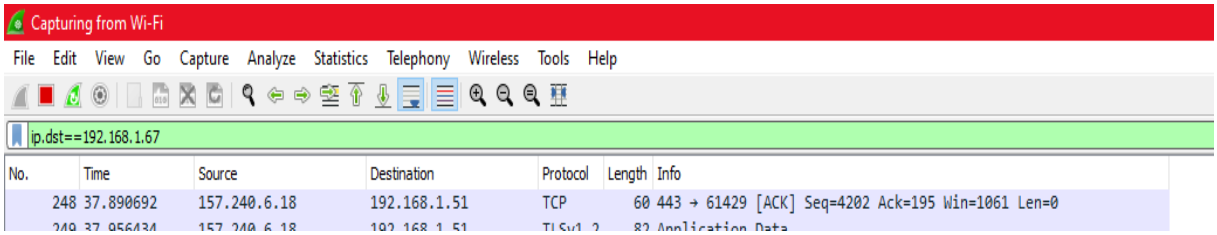


Figura 4-21.: Filtro de IP destino y puerto en WIRESHARK, **Fuente:** Elaboración propia

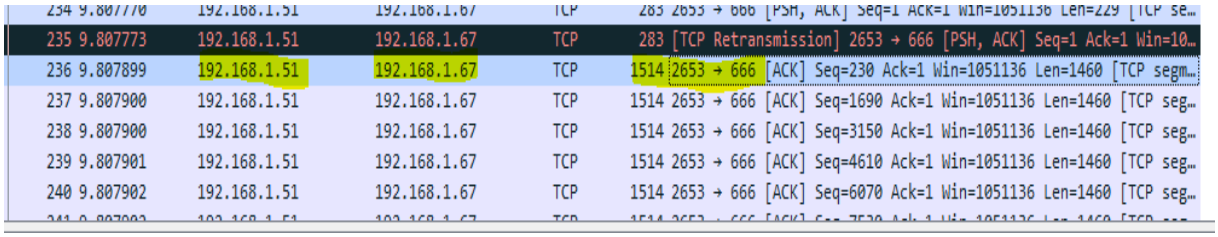


Figura 4-22.: Paquetes de WIRESHARK que contienen el mensaje enviado, **Fuente:** Elaboración propia

Al momento de analizar los paquetes encontrados se pudo comprobar que el mensaje viaja en texto claro con todo el contenido confidencial de los pacientes, dejando un gran agujero de seguridad en el envío de estos archivos. El contenido de los paquetes se puede apreciar en la Figura **4-23**.

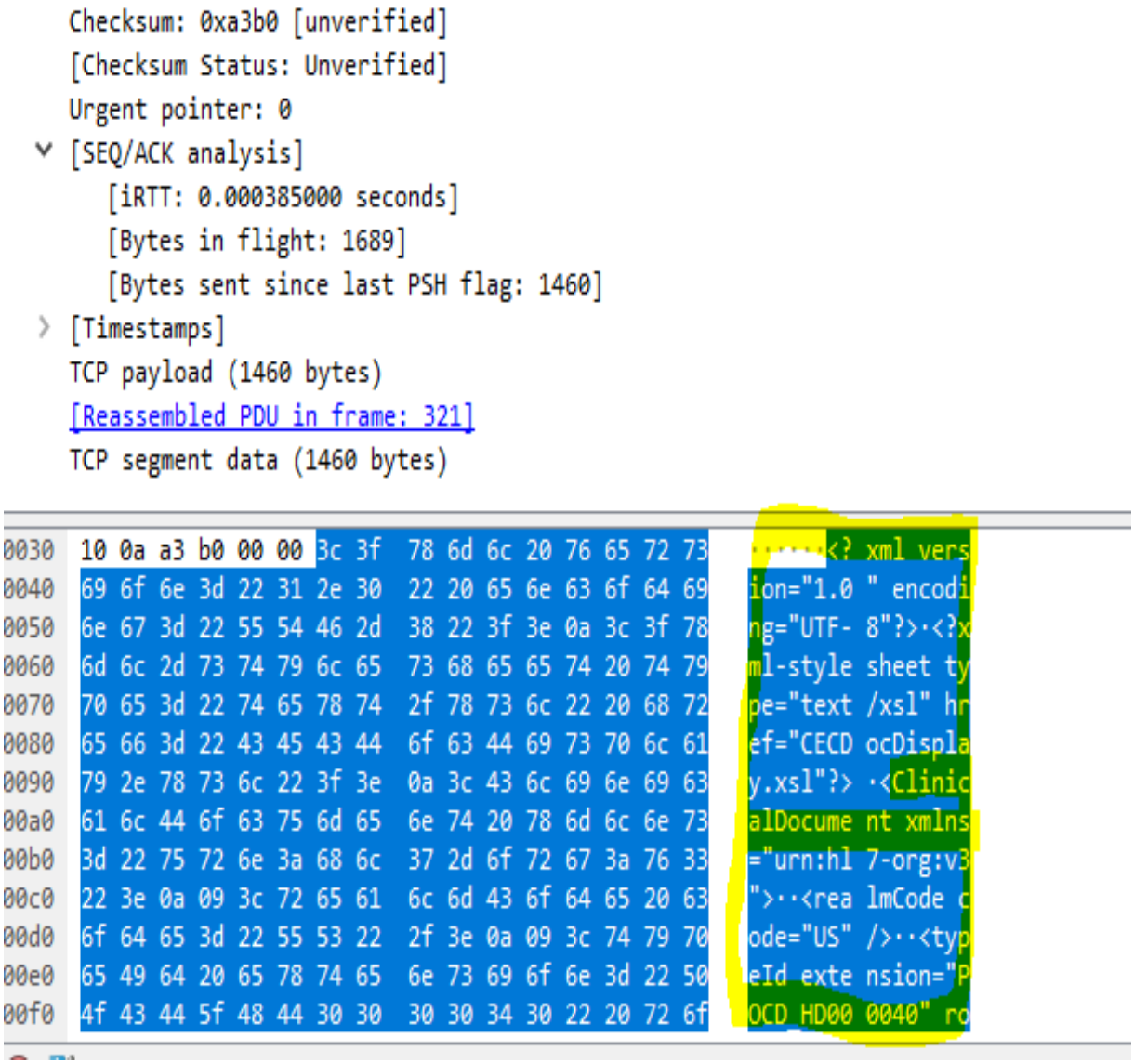


Figura 4-23.: Contenido de los Paquetes de WIRESHARK en ip destino, **Fuente:** Elaboración propia

ATAQUE DE Cross-site scripting O XSS

Las investigaciones desarrolladas para el año 2020 dieron por resultado que la vulnerabilidad para inyectar código java script en la plantilla CDA.xsl fue mitigada al cambiar las cabeceras. En la antigua plantilla se encontraba una vulnerabilidad que daba cabida al atacante para que realice un XSS que permita la ejecución de código arbitrario de javascript dentro del visor C-CDA, un atacante podría robar las cookies del navegador y publicarlas nuevamente en un servidor externo.

Una oportunidad para los ataques de inyección es el manejo predeterminado de un nonXML-Body CDA catalogado en la CVE como CVE-2014-3861[41]. Figura **4-24** Esto significa que

un atacante puede ejecutar JavaScript arbitrario al proporcionar una referencia como la Figura **4-25**:

```
<xsl:template match='n1:component/n1:nonXMLBody'>
  <xsl:choose>
  <!-- if there is a reference, use that in an IFRAME -->
  <xsl:when test='n1:text/n1:reference'>
  <IFRAME name='nonXMLBody' id='nonXMLBody' WIDTH='80%' HEIGHT='600' src='{n1:text/n1:reference/@value}' />
</xsl:when>
```

Figura 4-24.: manejo predeterminado de un nonXMLBody CDA, **Fuente:** [41]

```
<nonXMLBody>
  <text>
    <reference value="javascript:alert(parent.document.cookie);"/>
  </text>
</nonXMLBody>
```

Figura 4-25.: Código JavaScript, **Fuente:** [41]

La representación HTML de salida XSLT incluiría el siguiente fragmento peligroso Figura **4-26**:

```
  <iframe src="javascript:alert(parent.document.cookie);"></iframe>
```

Figura 4-26.: Código incrustado en el IFRAME de alerta, **Fuente:** [41]

La investigación del ataque XSS fue realizado por el Programa de Informática de Salud Computacional, Boston Children's Hospital, Boston, MA, con la empresa SMART titulada Security vulnerabilities in C-CDA Display using CDA.xsl [41]. Este ataque en este momento YA NO ES EFECTIVO PORQUE SE ENCUENTRA MITIGADA LA VULNERABILIDAD CVE-2014-3861 al cambiar su plantilla CDA.xsl.

TERCERA FASE

4.2.1. Evaluar el Prototipo de Framework Propuesto.

INSTALACIÓN Y EJECUCIÓN DEL PROTOTIPO DE FRAMEWORK

Para este paso se realizó una investigación exhaustiva mencionada en Apartado 4.1.1, donde se escogieron las mejores herramientas OPEN SOURCE para la mitigación de las vulnerabilidades. Se instaló la herramienta OSSEC IDS Anexo C Figura **4-27**.

Figura 4-27.: Logotipo de la Herramienta OSSEC, **Fuente:** OSSEC

Y la herramienta Mod Security IPS Anexo E Figura **4-28**.

Figura 4-28.: Logotipo de la Herramienta Mod Security, **Fuente:** Mod Security

En esta tesis se investigó y probó que las dos herramientas funcionan perfectamente en conjunto, prueba de ello fue el realizar una conferencia en el 15 festival de instalación de software libre FLISOL llevado a cabo en el Instituto tecnológico metropolitano de Medellín en el año 2019 donde se probó que las dos herramientas trabajan sin ningún problema y bajo recursos básicos en un servidor linux, por lo tanto, se unificaron para crear una máquina virtual con las herramientas actualizadas y configuradas previamente, creando un visor web con el apoyo de las herramientas Mod security y la plataforma de OSSEC donde se mostraran los ataques detectados por OSSEC y la prevención que Mod Security da en estos casos, con su regla de baneo de IP indefinidamente.
La prueba bajo ambiente controlado se realiza de la misma forma que en el Apartado 4.2 pero activando el prototipo de Framework, primeramente, activamos OSSEC reactivando el servicio de apache 2 con el comando **systemctl restart apache2** como se muestra en la Figura **4-29**.

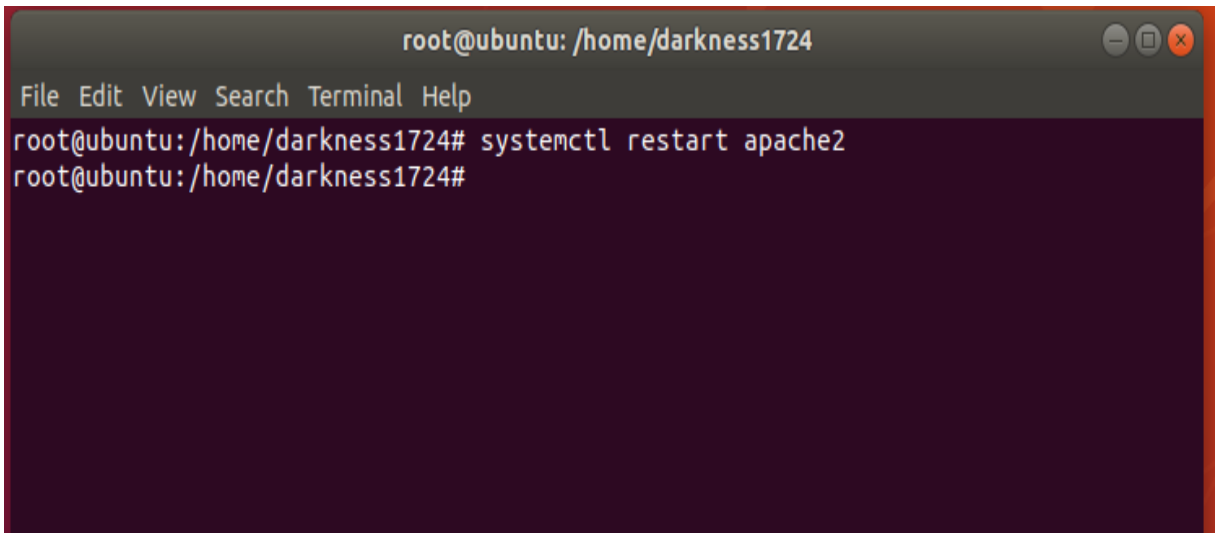


Figura 4-29.: Reinicio de Servicio apache2, **Fuente:** Elaboración propia

Activamos Mod Security configurando el archivo modsecurity.conf el cual se encuentra ubicado en /etc/modsecurity/modsecurity.conf, para modificar el archivo usamos la sentencia gedit, para activar la herramienta se elimina el símbolo Sharp de SecRuleEngine y se cambia el Off por el On y al finalizar todo se reinicia el apache 2 con el comando service apache2 restart, como se muestra en las siguientes figuras, Figura **4-30**, Figura **4-31**, Figura **4-32**.

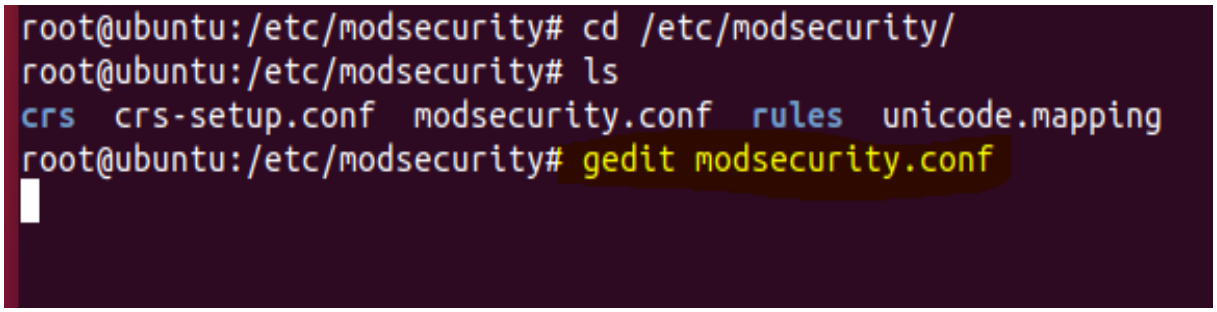


Figura 4-30.: Comando para Editar el archivo modsecurity.conf, **Fuente:** Elaboración propia

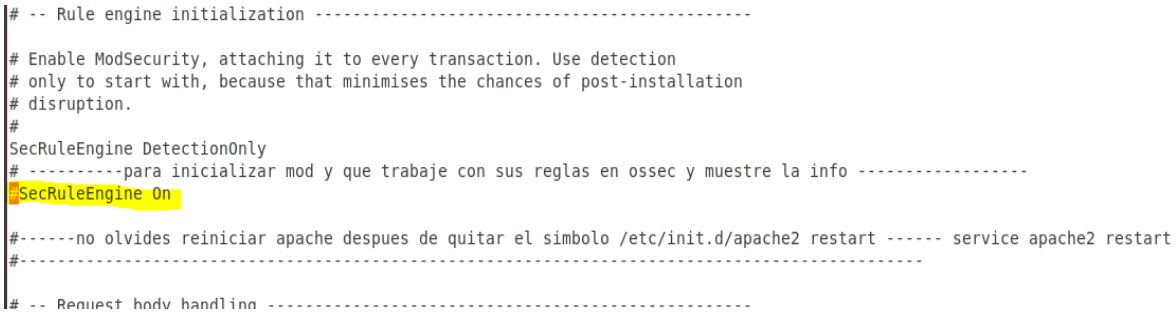


Figura 4-31.: Edición del Archivo modsecurity.conf, **Fuente:** Elaboración propia

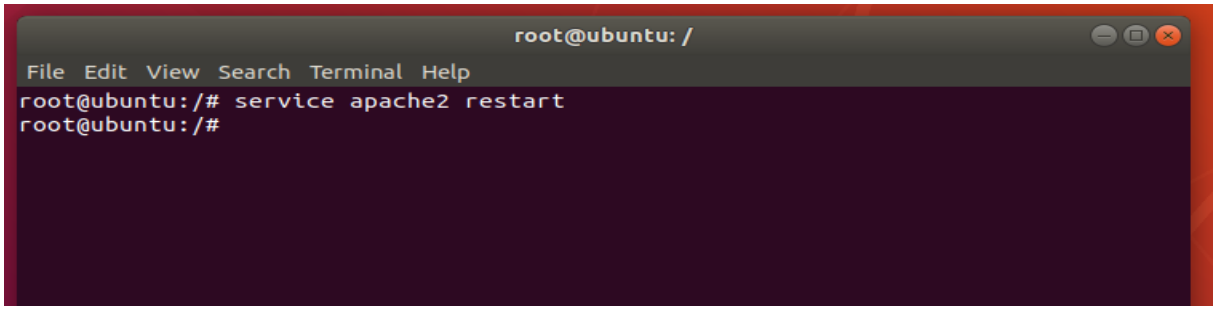


Figura 4-32.: Reinicio de apache2, **Fuente:** Elaboración propia

Ya activado el Mod Security y el Ossec se realizó nuevamente el envío del Historial Médico electrónico como se muestra en la Figura **4-33**, y se analizó los paquetes capturados en la herramienta Wireshark.

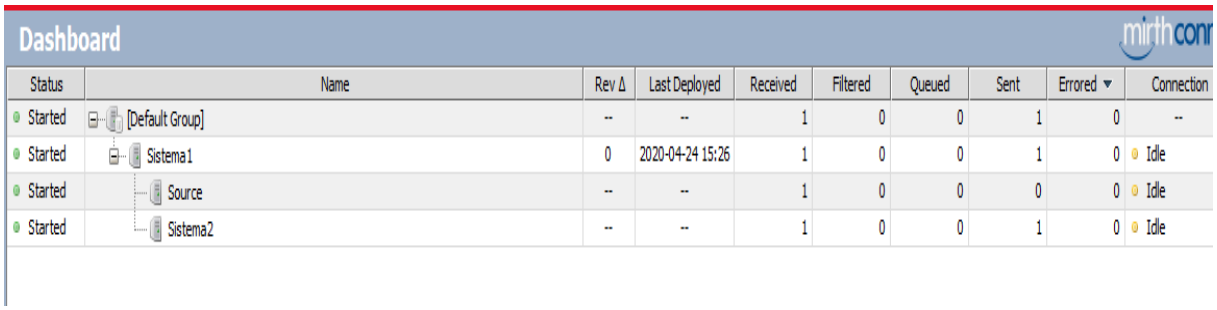


Figura 4-33.: Envío del archivo HL7 CDA de forma correcta, **Fuente:** Elaboración propia

Los paquetes analizados en Wireshark cambiaron de forma que ya no se puede visualizar el contenido de la plantilla CDA.xsl con los datos confidenciales, el resultado que existe en wireshark se nota una conexión rst y unas conexiones out of order de sincronización como se muestra en las figuras, Figura **4-34**, Figura **4-35**, Figura **4-36**.

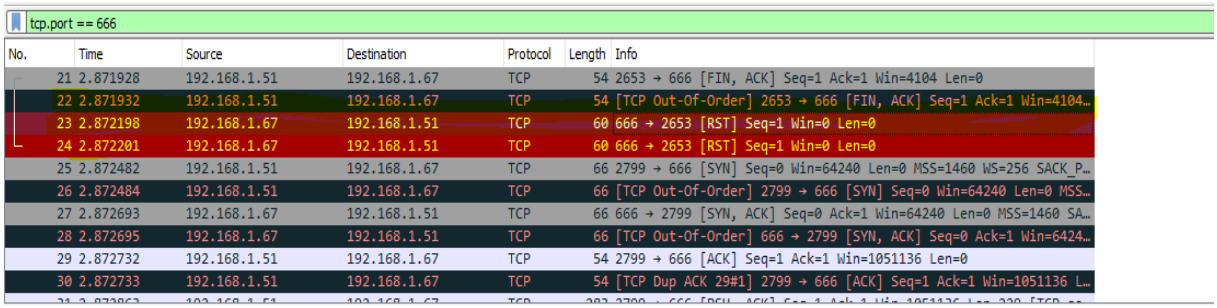


Figura 4-34.: Paquetes de Wireshark RST, **Fuente:** Elaboración propia

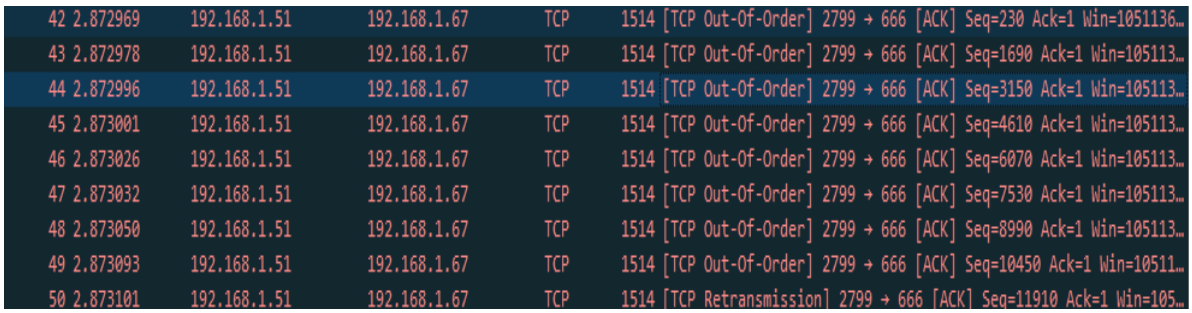


Figura 4-35.: Paquetes de Wireshark Out of Order **Fuente:** Elaboración propia

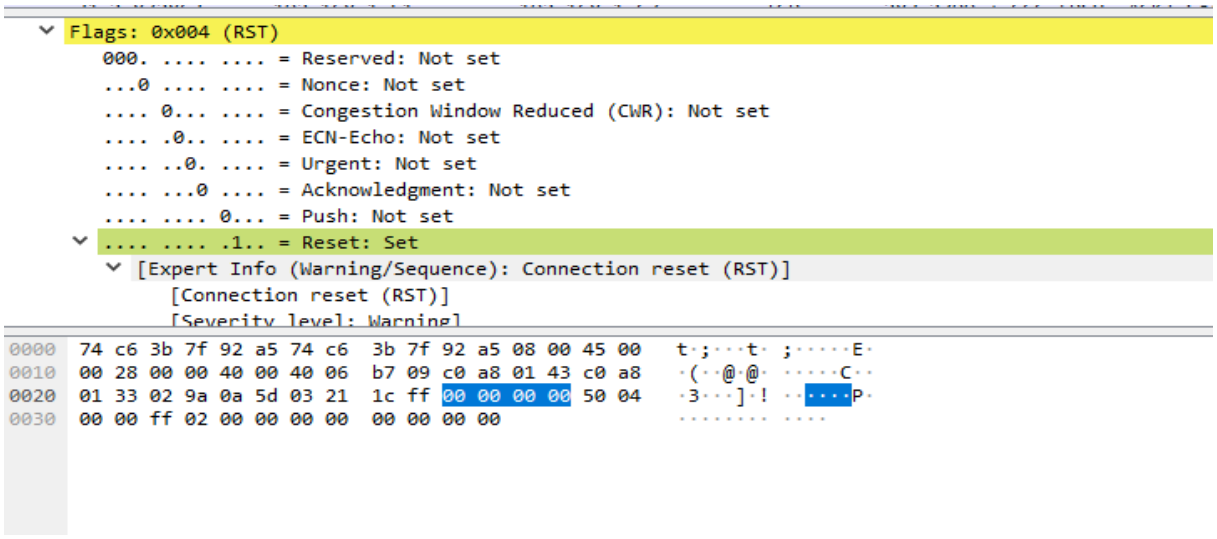


Figura 4-36.: Paquete de Wireshark Sin Lectura **Fuente:** Elaboración propia

Cuando se realizó un análisis de vulnerabilidades con la Herramienta Acunetix, esta lo que hace es forzar la aplicación web por medio de ataques múltiples para comprobar que tan segura es, en la figura Figura **4-37** se puede observar el funcionamiento del prototipo de framework.

Figura 4-37.: Framework de OSSEC y MOD SECURITY funcionando Exitosamente **Fuente:** Elaboración propia

En el momento que se Activó Acunetix en un escaneo rápido, actuó el prototipo de framework y bloqueo el acceso al ataque de acunetix tumbando la página del prototipo de framework modsec (Unión de Mod Security y Ossec) cuando se la abrió de forma remota, como se puede ver en la Figura **4-38**, pero de forma local la página del prototipo sigue funcionando y mostró el ataque y el bloqueo que realizó el Prototipo de Framework MODSEC como se muestra

en la Figura **4-39**. Al terminar el escaneo de Acunetix dio por resultado 0 vulnerabilidades, para poder ver los informes de acunetix ver Anexo F.

PARA DESCARGAR EL PROTOTIPO DE FRAMEWORK VER Anexo G.

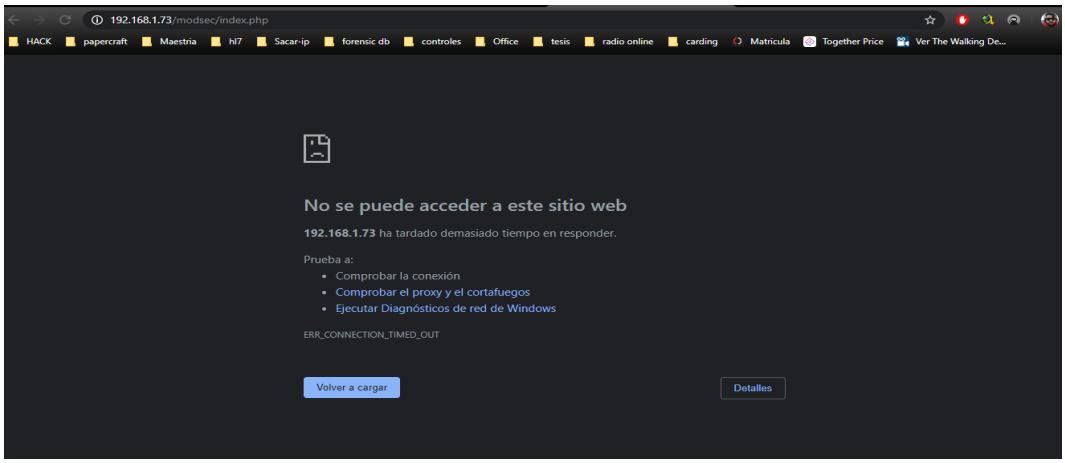


Figura 4-38.: Página remota del Prototipo de Framework Caída **Fuente:** Elaboración propia

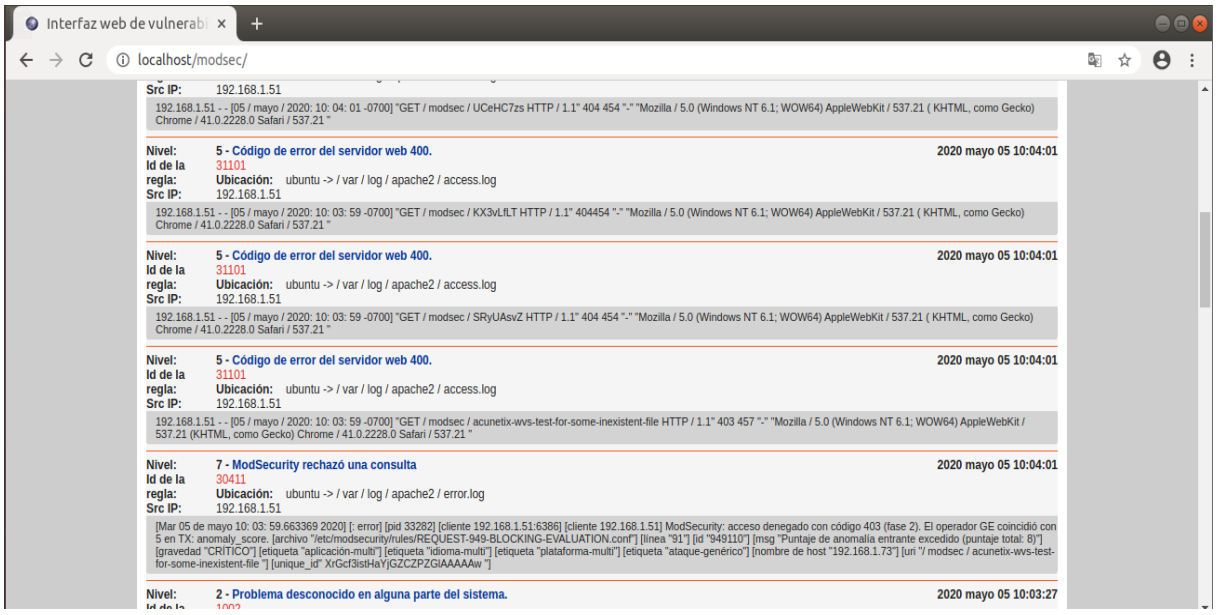


Figura 4-39.: Resultado efectivo de la mitigación y detección del ataque **Fuente:** Elaboración propia

5. Conclusiones y recomendaciones

5.1. Conclusiones

El trabajo de Investigación se desarrolló bajo la metodología propuesta dividida en 3 fases, la primera fase dio cumplimiento a los objetivos 1 y 2 los cuales son Identificar y caracterizar los diferentes problemas de confidencialidad que se presentan en el estándar HL7 CDA R2 y determinar cuál es la herramienta idónea para la detección y prevención que afecte la confidencialidad en el estándar HL7 CDA R2. Estos dos objetivos de la fase 1 fueron cumplidos a cabalidad al poder identificar y caracterizar los problemas de confidencialidad que se encontraron los cuales fueron la obtención de Información y la inyección de código malicioso el cual en este año 2020 ya fue subsanada, así mismo se dio cumplimiento exitoso al segundo objetivo al determinar las 2 mejores herramientas para detectar y prevenir los ataques que pudieran ser efectivos para obtener la información de los documentos HL7 CDA R2. Se pudo concluir de esta primera fase que el estándar HL7 CDA R2 si posee problemas de confidencialidad pero que pueden ser mitigadas de una forma eficiente con las herramientas idóneas, según el caso lo amerite, para esta investigación fueron efectivas las herramientas Mod Security como IPS y OSSEC como IDS las cuales fueron integradas para que trabajen de forma unificada y eficaz y así poder mitigar de forma eficiente los problemas de confidencialidad.

La segunda fase da cumplimiento al objetivo número 3 el cual es Implementar el prototipo de Framework de seguridad para preservar la confidencialidad de los datos en el estándar HL7 CDA R2, el desarrollo de este objetivo se basó en la instalación y configuración de las herramientas para que puedan funcionar de una manera unificada se puede concluir que las dos herramientas en conjunto hacen un trabajo completo de detección y prevención de ataques de forma muy efectiva y de muy bajo presupuesto ya que se está trabajando con herramientas open source las cuales podemos modificar a conveniencia según la empresa, el prototipo de Framework funciona de una manera eficaz en empresas pequeñas y medianas, para esta investigación se acomoda de manera efectiva ya que seria 2 entidades de salud trabajando en envíos de historiales médicos electrónicos.
El prototipo de Framework no necesita muchos recursos para su instalación, después de que ya se tiene instalado es de muy fácil manejo para los administradores de red y así poder reportar y reaccionar ante un incidente de seguridad de manera ágil y rápida ya que la detección y prevención es en tiempo real, este prototipo de framework no solo trabaja miti-

gando el envío de historiales médicos de forma segura, también mitiga y detecta los ataques más conocidos mostrándolos en el panel bajo un nivel de criticidad que se trabaja en la CVE Vulnerabilities, de esta forma se puede realizar análisis de riesgos ya contando con niveles de criticidad internacionales, al integrar Mod security al prototipo de Framework se está asegurando que el atacante o la máquina atacante va a ser baneada por medio de su IP por el tiempo que el administrador de red lo disponga, trabajan bajo reglas ya pre-diseñadas y probadas open source las cuales se pueden modificar a conveniencia de la empresa.

Y por último, pero no menos importante la tercera Fase da cumplimiento al objetivo número 4 el cual es, Evaluar la implementación del prototipo de Framework propuesto bajo pruebas de ambiente controlado para comprobar que los riesgos han sido mitigados. Para dar cumplimiento óptimo a este objetivo y culminar exitosamente con esta fase se trabajó en el mismo ambiente controlado en el cual se probaron las herramientas y el prototipo de framework, los resultados obtenidos fueron más que favorables al presentar la mitigación exitosa de la vulnerabilidad Man in the Middle de obtención de datos confidenciales de los historiales médicos HL7 CDA R2, es por esto que se realizó un análisis de vulnerabilidades con la herramienta ACUNETIX en su versión de prueba, pero esta herramienta sirvió para comprobar que el prototipo de framework trabaja de una manera ágil y efectiva al mostrar los ataques que trataba de hacer esta herramienta y al bloquear el acceso desde la IP atacante hacia el TARGET o víctima.

Al cumplir con estas tres fases satisfactoriamente se da por terminado y completado de forma efectiva el objetivo General que es Desarrollar un Prototipo de Framework para la mitigación de riesgos en la confidencialidad de los datos del estándar HL7 CDA R2, con el análisis de vulnerabilidades y el análisis de los paquetes del sniffer Wireshark se pudo comprobar el buen funcionamiento de este prototipo y que el desarrollo de este para mitigar los riesgos de confidencialidad para el estándar HL7 CDA R2 fue exitosa.

5.2. Recomendaciones

Para tener en cuenta, la principal recomendación para futuras investigaciones es trabajar con herramientas que se encuentren en constante investigación y actualización de 0 days para mitigar al máximo los problemas de confidencialidad en las entidades médicas a las cuales se las ha visto bastante desatendidas en la parte de seguridad de la información, para algunos entes la información que se envía en estos historiales médicos no son de carácter confidencial, pero la información es totalmente privada del paciente y sin su autorización la información personal de este debe ser protegida de una manera adecuada y eficiente.

Se podría trabajar en un sistema de envío y recibo de Historiales Médicos estandarizado en Colombia de plataforma interactiva y amigable para sus usuarios con un nivel alto de

interoperabilidad el cual ya está estipulado en la ley relativamente nueva LEY 2015 DEL 31 DE ENERO DE 2020 la cual nos habla de regular la interoperabilidad de las historias clínicas electrónicas respetando e Hábeas data y la reserva de la misma, obligando a estas entidades clínicas o médicas a diligenciar y disponer los datos, documentos y expedientes de la historia clínica en la plataforma de interoperabilidad que le disponga el Gobierno nacional.

Se recomienda como un desafió a trabajar , en una investigación de implementación de estos módulos y unificarlos en diferentes sistemas operativos, ya que en esta tesis de maestría solo se trabajo con UBUNTU v. 18.

A. Anexo: Instalación MIRTH CONNECT

Para la instalación de este motor de envíos se descarga el instalador desde la página [1], se procede a descargar el JDK (JRE) desde la página de Oracle [2]. Cuando ya se tiene instalado el JRE ya se puede proceder a la instalación normal del motor presionando en siguiente como lo muestra la Figura **A-1**.

Figura A-1.: Instalación Motor MIRTH CONNECT 1, **Fuente:** Elaboración propia

Después de la Instalación se procede abrir el motor Mirth connect server manager donde se instalará por sí solo y se abrirá la pantalla de configuración del servidor del motor de envíos como lo muestra la Figura **A-2** y se presiona el botón launch.

[1] https://www.caduceus.es/descargar-mirth-connect
[2] https://www.oracle.com/java/technologies/javase-jre8-downloads.html

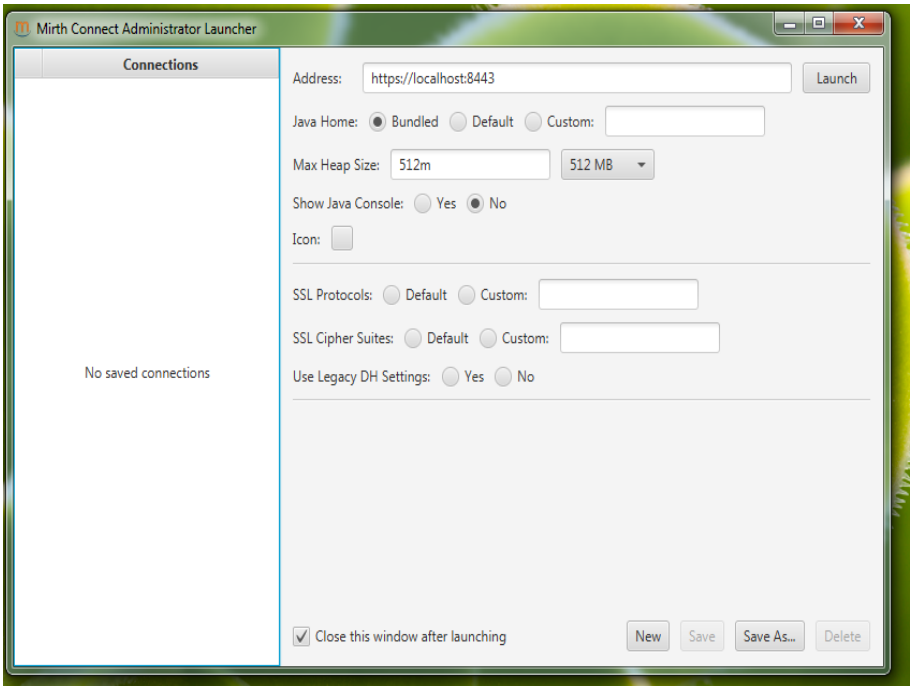


Figura A-2.: Instalación Motor MIRTH CONNECT 2, **Fuente:** Elaboración propia

Cuando ya se obtiene acceso a la ventana de ingreso donde se pide usuario y contraseña, por definición sus credenciales son admin y contraseña admin en la Figura **A-3** se muestra el ingreso con las credenciales para ingresar al panel donde se configurará los canales de envío para poder realizar las pruebas respectivas en el Anexo B.

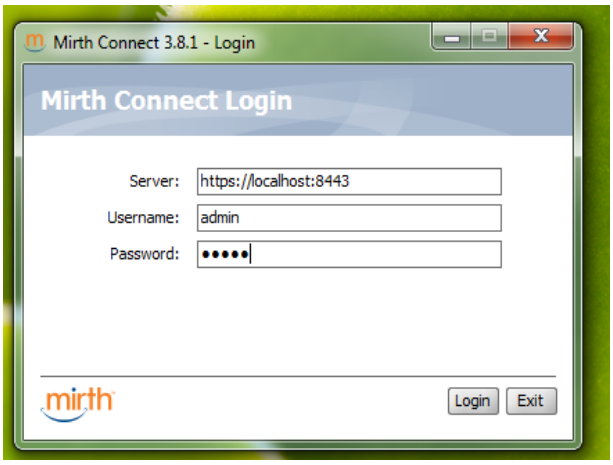


Figura A-3.: Instalación Motor MIRTH CONNECT 3, **Fuente:** Elaboración propia

B. Anexo: Configuración y Creación de canales en MIRTH CONNECT

Después de haber realizado la instalación correspondiente se abre el programa por primera vez y pide cambio de contraseña y algunos datos personales para iniciar como se muestra en la Figura **B-1**

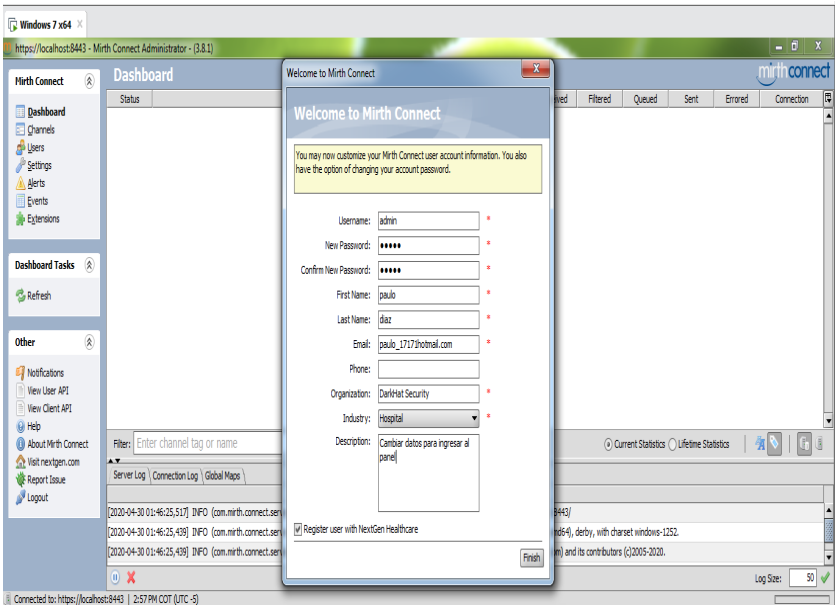


Figura B-1.: Cambio de Datos Personales Motor MIRTH CONNECT, **Fuente:** Elaboración propia

Cuando se han cambiado las credenciales se empieza por la creación de los canales presionando a la pestaña del lado izquierdo channel que está marcada con un círculo amarillo y después más abajo se activa la pestaña que señala la flecha roja NEW CHANNEL para poder crear el canal de transmisión y la entidad que recibe los historiales médicos, como lo muestra la Figura **B-2**.

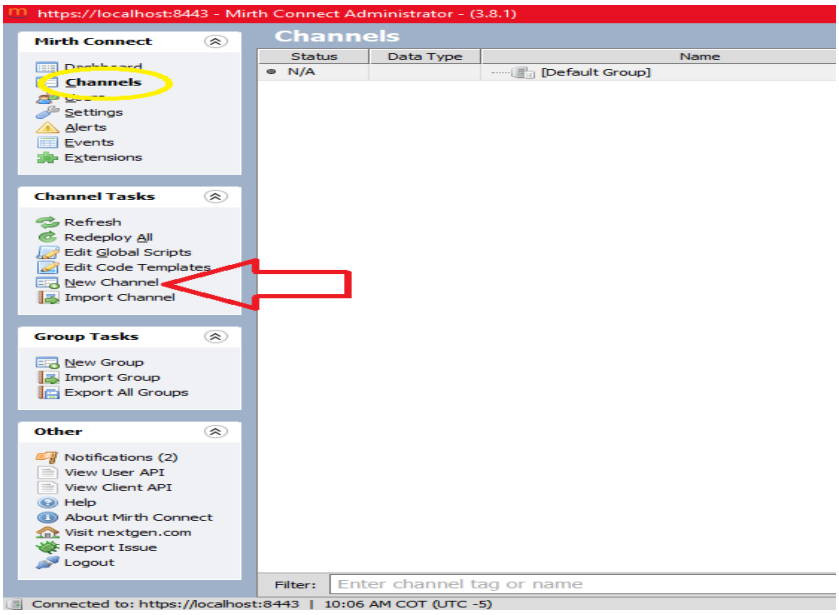


Figura B-2.: Creación de canales en Motor MIRTH CONNECT, **Fuente:** Elaboración propia

Después de haber abierto NEW CHANNEL se abre una interfaz en la pestaña Summary, donde se pondrá el nombre del canal que para este caso se llamará Sistema 1 el cual será la primera entidad hospitalaria (la que ENVÍA los historiales médicos), se muestra en la Figura **B-3**. Se procede a cambiar qué tipo de datos serán enviados en el botón Set Data Types y que datos se recibirán en la segunda entidad hospitalaria, para este caso se cambiarán xml en envío y cuando se reciba será igual en xml como lo muestra la Figura **B-4**.

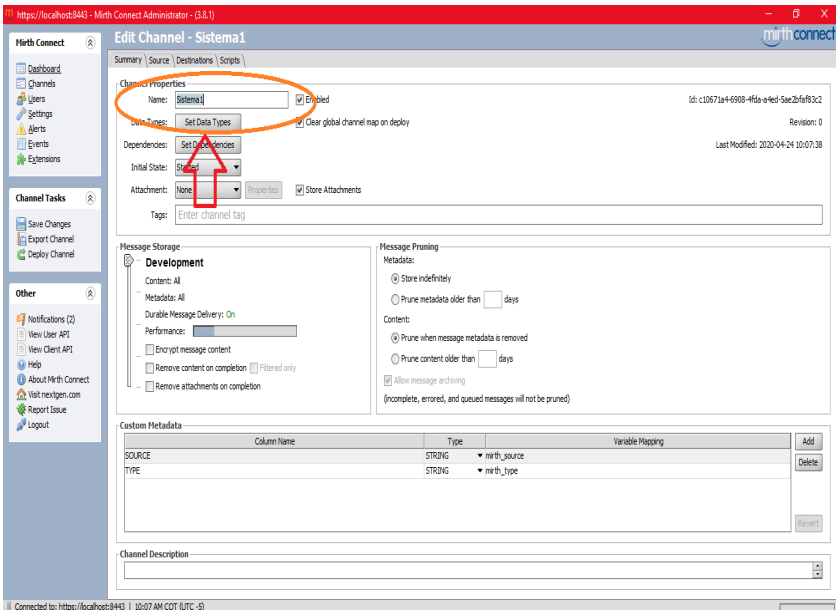


Figura B-3.: nombre del canal en Motor MIRTH CONNECT, **Fuente:** Elaboración propia

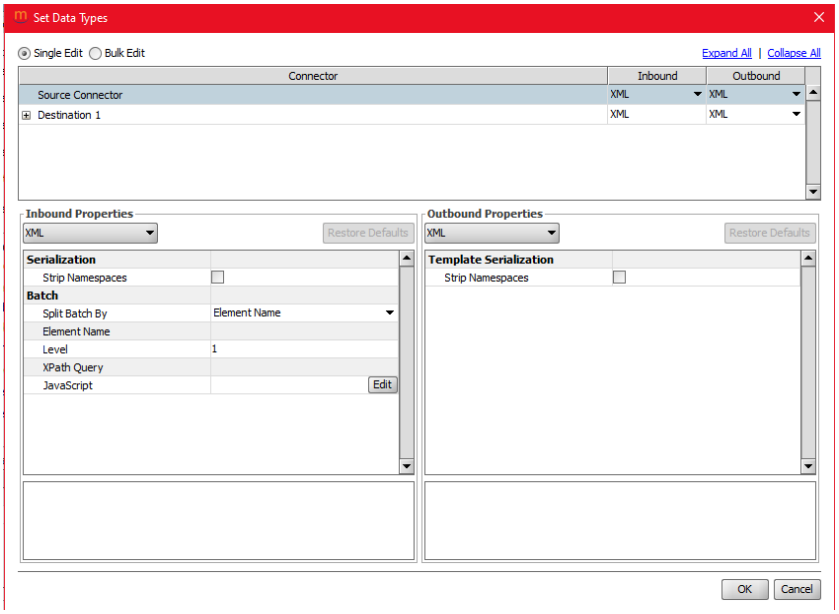


Figura B-4.: Cambio de Formato de archivos en Motor MIRTH CONNECT, **Fuente:** Elaboración propia

Se procede a ir a la pestaña Source donde se pondrá el tipo de conexión que se va a realizar y se escoge la frecuencia con la que estará enviando los archivos, para este caso se escoge File Reader como tipo de conexión y la frecuencia de envío será el intervalo de tiempo viene predefinido cada 5 segundos como se muestra en la Figura **B-5**.

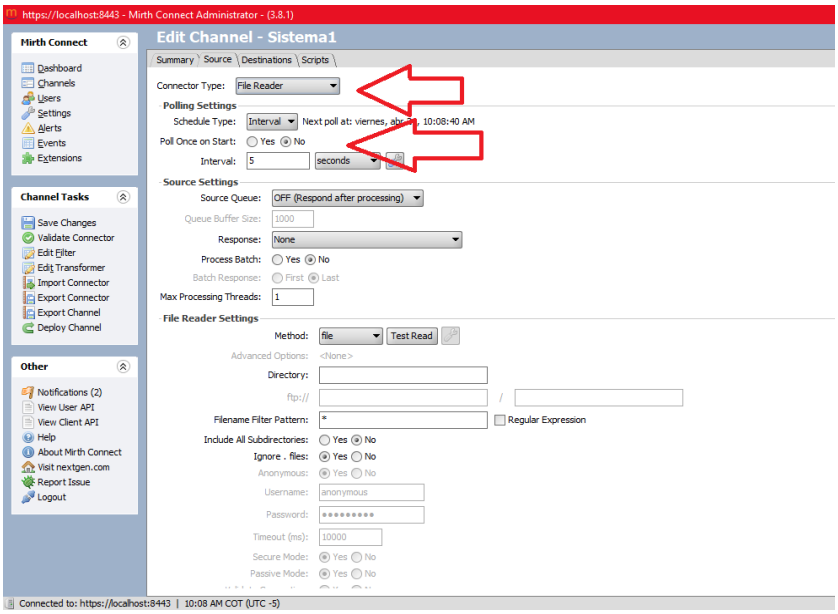


Figura B-5.: Tipo de Conexión e Intervalo de Envío en Motor MIRTH CONNECT, **Fuente:** Elaboración propia

Luego se define que el archivo después de que sea enviado se elimine de la carpeta donde estarán los historiales médicos CDA, y se escribe la dirección de esta carpeta contenedora de estos Historiales en la sección Directory para armar la conexión como se muestra en Figura **B-6**.

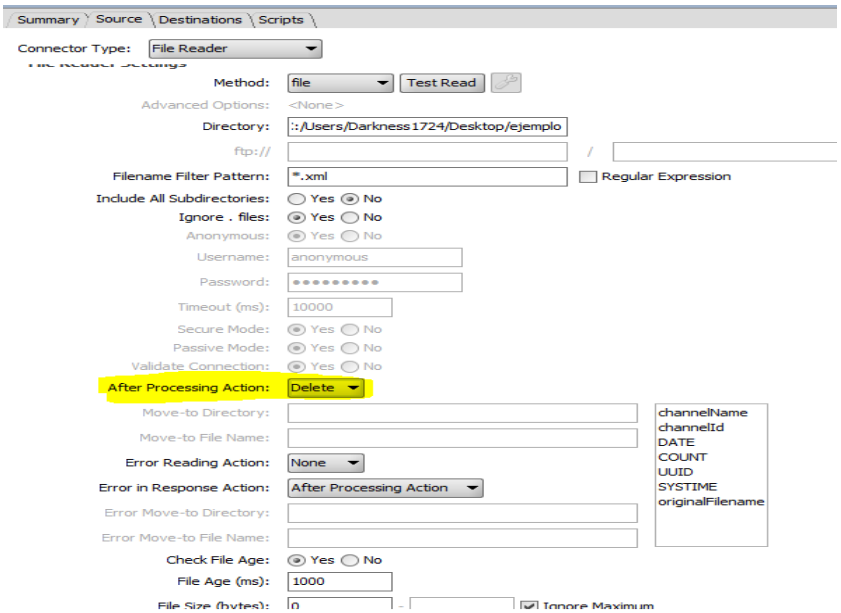


Figura B-6.: Eliminando Archivo enviado y conexión con directorio contenedor, **Fuente:** Elaboración propia

Después se dispone a cambiar el tipo de archivo en FILE, se escribe la dirección de la carpeta que contiene los Historiales médicos electrónicos HL7 CDA R2 en DIRECTORY, en este caso es una carpeta contenida en el escritorio llamada Ejemplo, se filtra el tipo de archivo que será enviado con la función *.xml como se muestra en la Figura **B-7**. por último se realiza el test que la conexión se encuentre en línea y correcta presionando el botón TEST READ, vea la Figura **B-8**.

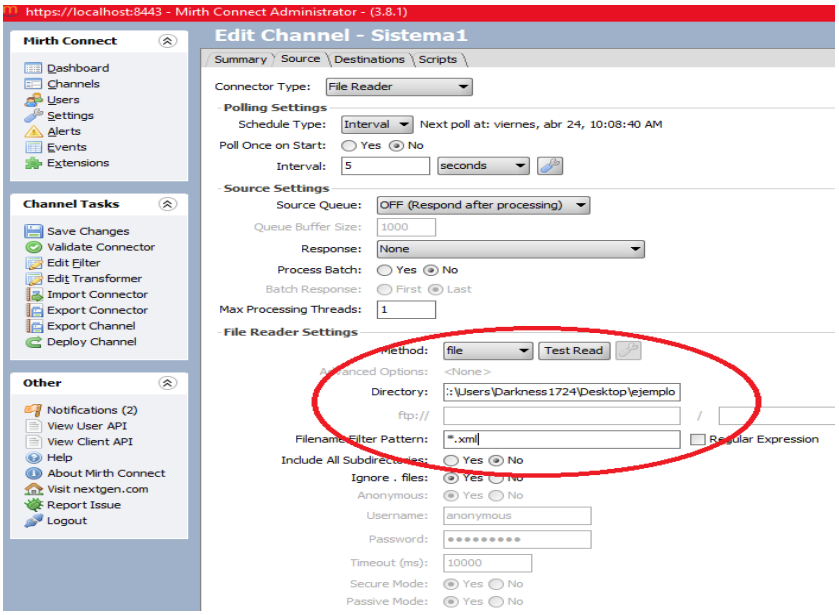


Figura B-7.: Dirección de carpeta contenedora de CDA y filtro de archivos XML, **Fuente:** Elaboración propia

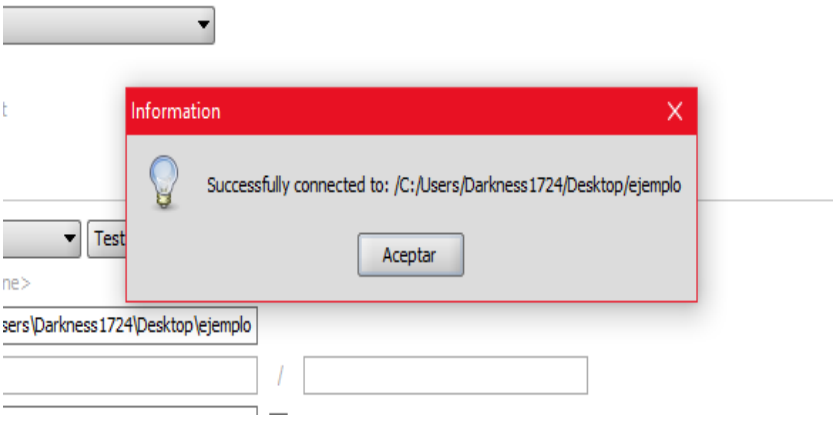


Figura B-8.: Testeo de Conexión Correcta, **Fuente:** Elaboración propia

Ya por últimos pasos se debe configurar la entidad Receptora de los CDA y esto se realiza en la pestaña DESTINATION, en donde se pondrá nombre a la entidad Hospitalaria que va

a recibir los CDA que en este caso es Sistema2, se configura el tipo de conexión como File Writer para que el archivo enviado sea escrito en la carpeta que recibe los CDA, la sección es Directory en donde se escribe la dirección de la carpeta que en esta investigación es la carpeta del programa visor de archivos CDA Backbeach Software que también se configura en esta sección, para que el archivo enviado tenga el mismo nombre y formato xml se configura una excreción predefinida Message ID la cual solo se arrastra hacia la sección file name como se puede apreciar en la Figura **B-9**.

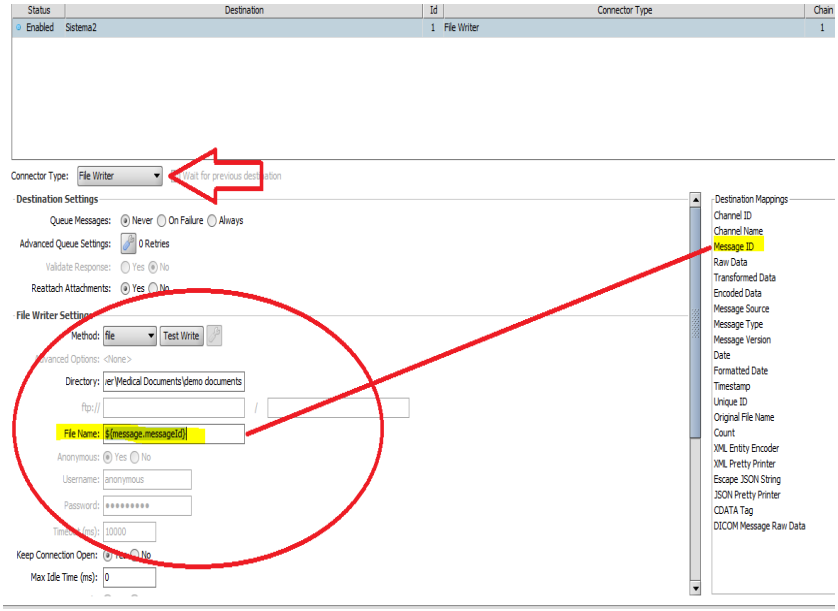


Figura B-9.: Configuración Destino de archivos CDA (Entidad Hospitalaria Receptora, **Fuente:** Elaboración propia

Para la configuración de la plantilla, en esta misma sección se configura en la parte de template, en donde se pegara el contenido de la plantilla CDA.xsl y en las secciones como el nombre, como el id del canal y transformaciones de datos se arrastra y se pegan estas variables que se encuentran de lado derecho, donde llevaran la información de los pacientes y se cambiarán por las variables quedando así el documento estructurado, listo para guardar y testear las conexiones como lo muestra esta secuencia de figuras, Figura **B-10**, Figura **B-11**, Figura **B-12**, Figura **B-13**.

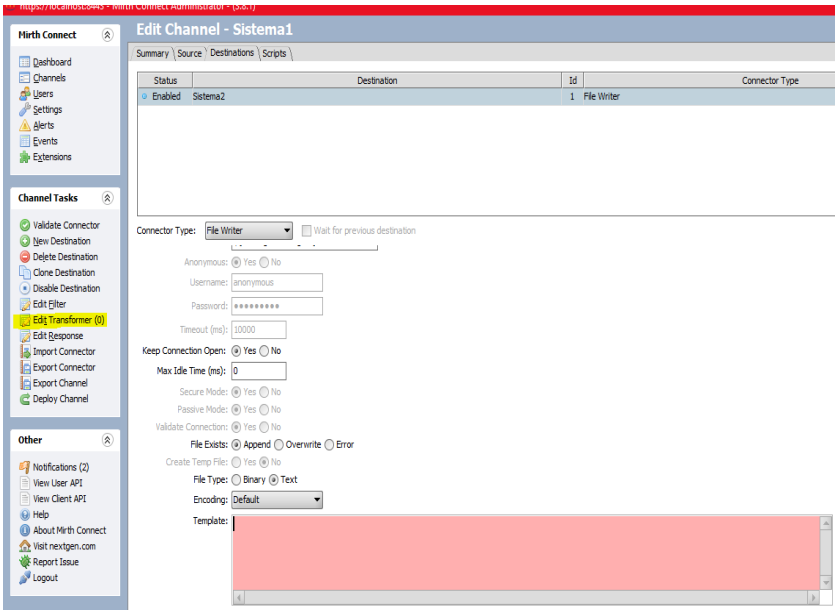


Figura B-10.: Editar transformación de la plantilla CDA.xsl, **Fuente:** Elaboración propia

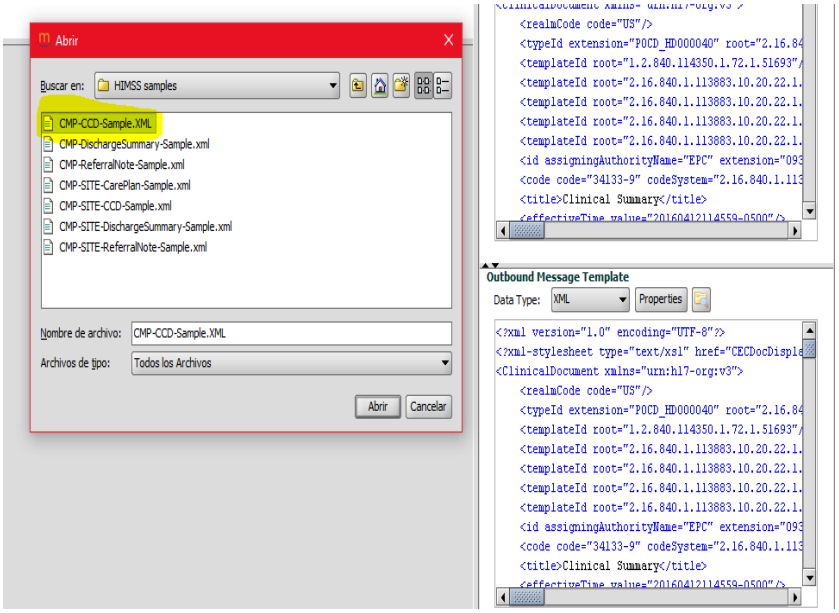


Figura B-11.: Abriendo plantilla de edición para que sea transformada en versión xml editable, **Fuente:** Elaboración propia

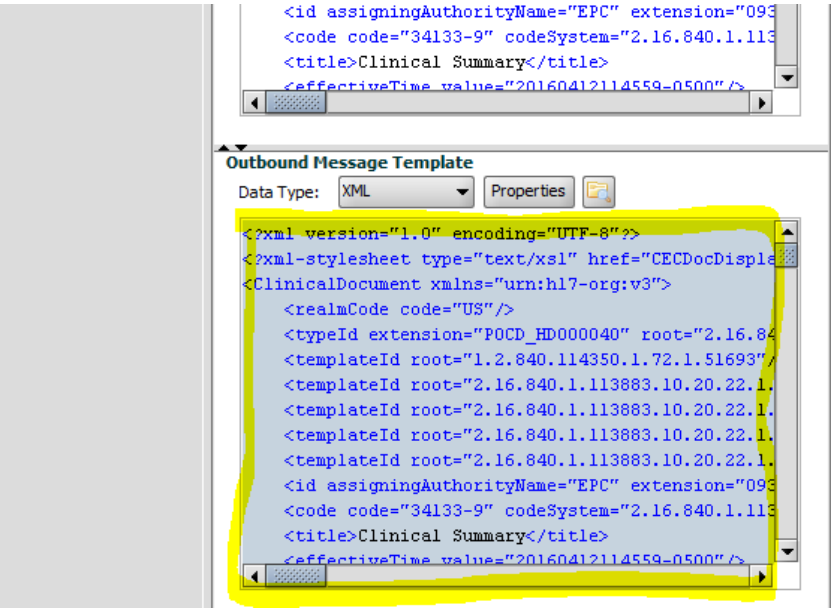


Figura B-12.: Abriendo plantilla de edición para que sea transformada en versión xml editable, **Fuente:** Elaboración propia

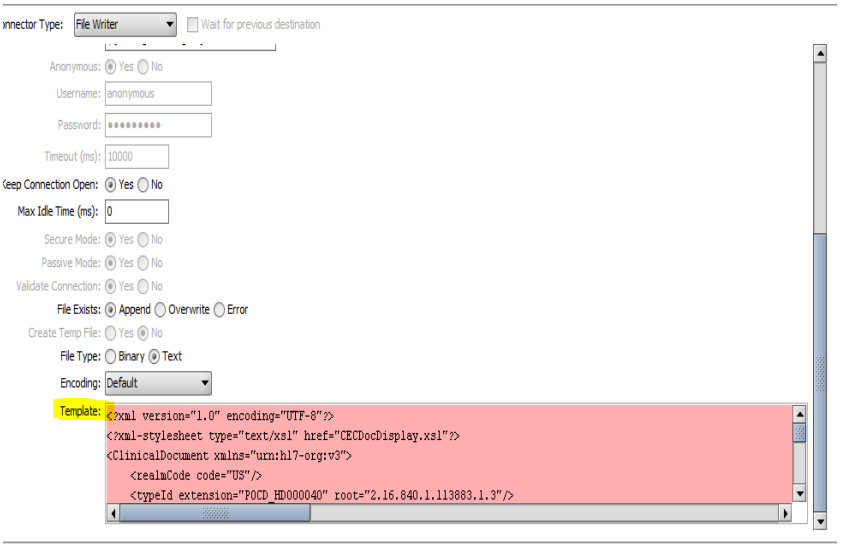


Figura B-13.: Plantilla editada terminada, **Fuente:** Elaboración propia

Al finalizar la edición de la plantilla CDA.xsl se procede a guardar los cambios realizados en todo el canal de entidad hospitalaria que envía los CDA y la entidad receptora de estos CDA y se valida la conexión completa de este canal entre Sistema 1 a Sistema 2 como se puede apreciar en las siguientes figuras, Figura **E-1**, Figura **E-2**.

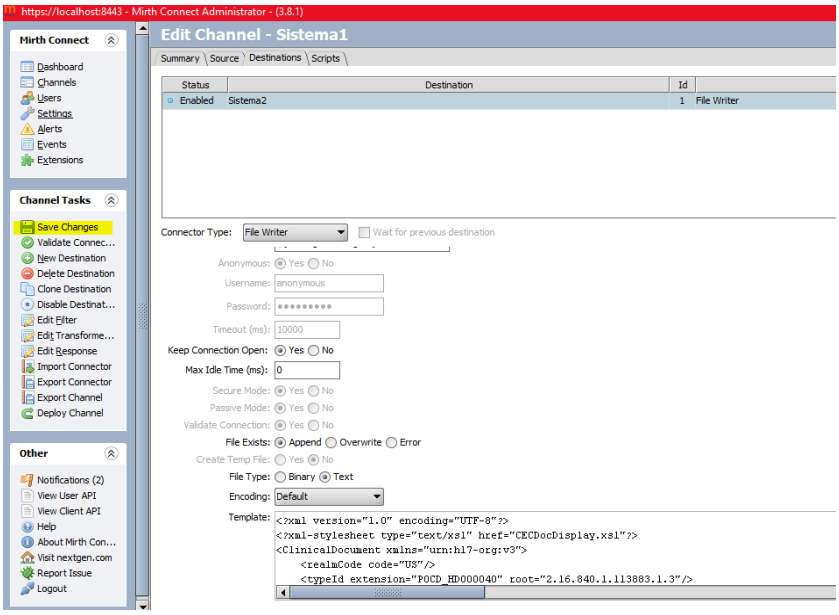


Figura B-14.: Guardado de Canal y Receptor, **Fuente:** Elaboración propia

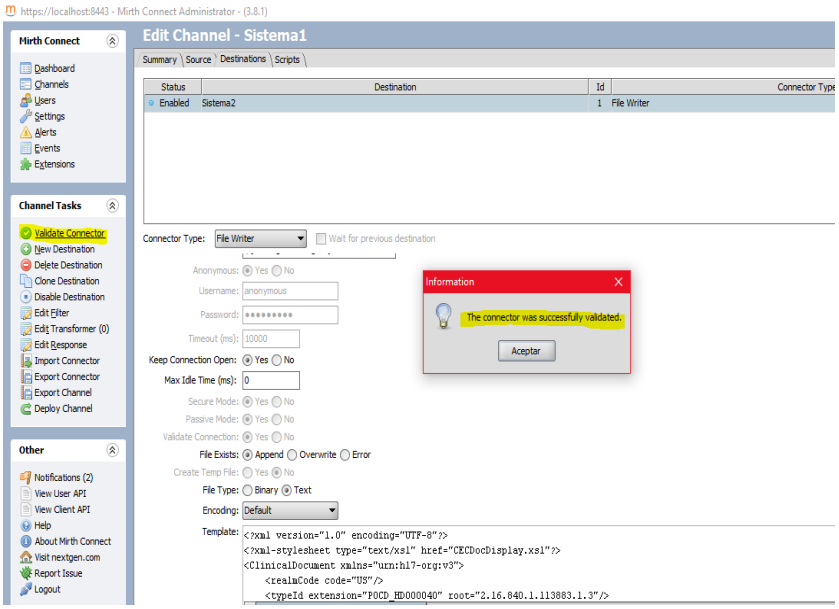


Figura B-15.: Conexión completa de este canal entre Sistema 1 a Sistema 2, **Fuente:** Elaboración propia

Por último, para que el funcionamiento del Canal sea efectivo y correcto se realiza la implementación al panel o Dashboard dando click en la opción Reploy All donde se abrirá una opción para preguntar si está seguro de implementar este canal para su funcionamiento, se da click en SI y el canal quedara implementado como se muestra en las Figuras, Figura **F-1**, Figura **G-1**.

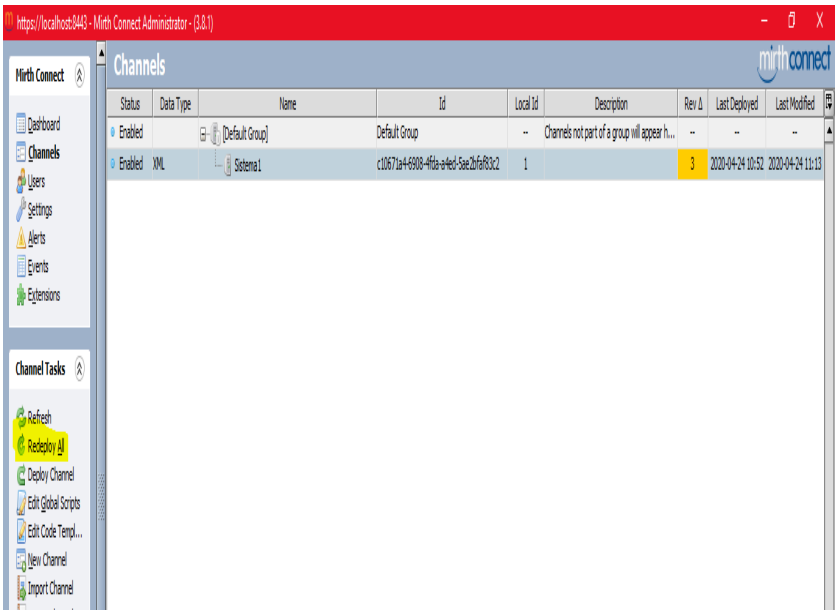


Figura B-16.: Opción Reploy All, **Fuente:** Elaboración propia

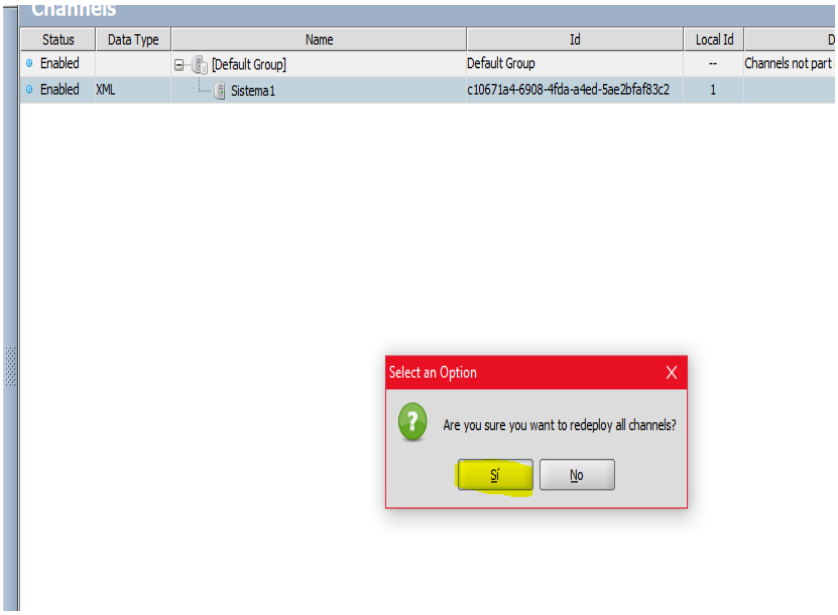


Figura B-17.: Aceptación del canal en funcionamiento, **Fuente:** Elaboración propia

C. Anexo: Plantilla de Archivo CDA.xsl

La plantilla CDA.xsl Fue subida a un gestor de descargas para más comodidad y por su larga extinción de caracteres, para descargarlo dirígete al link [1]. Esta plantilla es la más importante para la transformación de los datos HL7 a Historial Médico Electrónico y para visualizarlo en web por medio del Visor Backbeach Software como se ve en la siguiente Figura **C-1**.

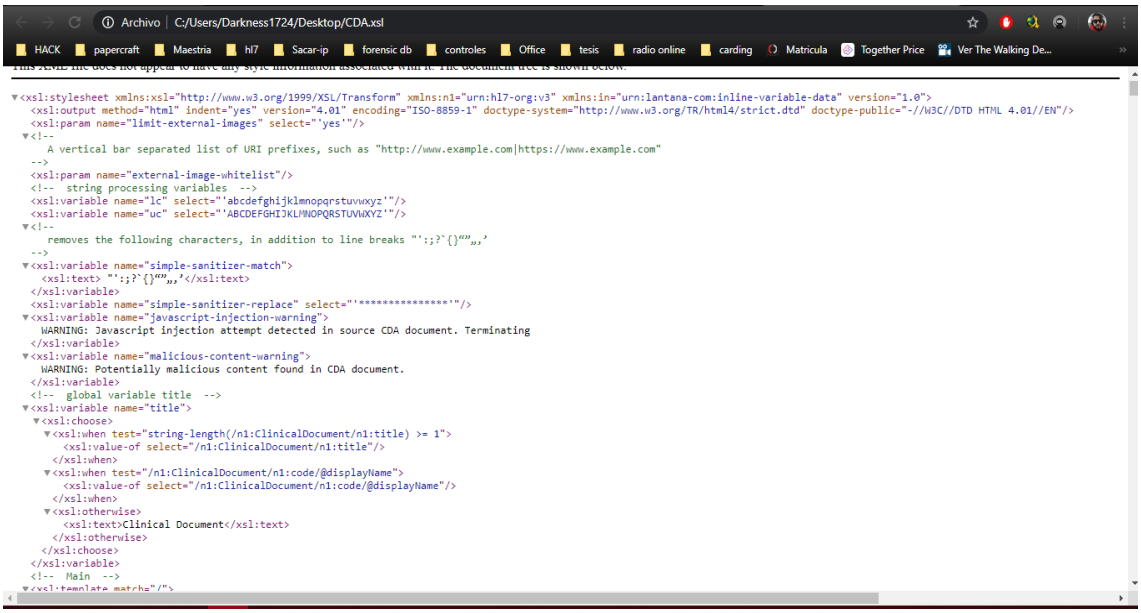


Figura C-1.: Plantilla de Archivo CDA.xsl, **Fuente:** Elaboración propia

[1] https://www.mediafire.com/file/6glt4dt9suiyj7t/CDA.xsl/file

D. Anexo: Instalación de Herramienta OSSEC IDS

La instalación se realiza por pasos:
El primer paso es estar en la consola y entrar como súper usuario con el comando sudo su xxxx, las (xxx) es su clave de acceso, ya al estar como súper usuario se empieza con los comandos de instalación:

- Se actualiza los sistemas operativos, apt-get update, apt-get upgrade.
- Instalar apache, **apt-get install apache2.**
- Instalar mysql server, **apt-get install mysql-server**.
- Instalar librería php, **apt-get install apache2 libapache2-mod-php**.
- Instalar librería Zlib, **apt-get install libz-dev**.
- Instalar paquete build-essentials, **apt-get install build-essential**.
- Descargar Ossec, **wget https://github.com/ossec/ossec-hids/archive/3.2.0.tar.gz**.
- Extraer archivo .tar.gz, **tar -zxvf 3.2.0.tar.gz**.
- instalar Ossec versión 3.2.0, ingresar a la ruta donde se halla descomprimido el archivo ossec-hids-3.2.0 y ejecutar: **./install.sh**.
- Escoger el idioma de instalación y presionar ENTER para continuar.
- Escoger tipo de instalación (servidor).
- Configurar variables de instalación, (presionar ENTER para dejarlo por defecto).
- configurar el sistema OSSEC HIDS, Desea recibir notificación por correo electrónico? (s/n): (usted elige sí o no, recomendable escoger ”s”).
- hemos encontrado su servidor de correo (dominio): (confirmar sí es correcto ”s.º no ”n”).

- Desea usted agregar el servidor de integridad del sistema? (s/n): (escoger ”s”).
- Desea usted agregar el sistema de detección de rootkit? (s/n): (escoger ”s”)
- Desea habilitar respuesta activa? (s/n): (escoger ”s”)
- Desea habilitar la respuesta desechar en el firewall? (s/n): (escoger ”s”)
- Desea usted agregar más IPs a la lista blanca? (s/n)?: (escoger a conveniencia, para este caso, escoger ”n”)
- Desea usted habilitar syslog remoto (puerto 514 udp)? (s/n): (escoger ”s”)
- Presionar ENTER para continuar.
- reiniciar el servicio de apache, **/var/ossec/bin/ossec-control restart**
- Descargar interfaz gráfica de OSSEC, **wget https://github.com/ossec/ossec-wui/archive/0.9.tar**
- Extraer archivo .tar.gz, **unzip 9.0.zip**
- Instalación de interfaz gráfica ossec, Movemos carpeta ossec-wui-0.9 a carpeta html y al mismo tiempo, cambiamos de nombre a la carpeta a .ºssec”,**mv ossec-wui-0.9 /var/www/html/ossec**.
- Ingresamos a la ruta, **cd /var/www/html/ossec**
- Instalamos la interfaz gráfica, **./install.sh**
- Registramos un usuario y una contraseña
- registramos el servidor web que se está usando, para este caso registramos **apache**
- reiniciamos servicio apache2, **systemctl restart apache2**
- Asignamos permisos a interfaz gráfica (ossec-wui) para poder leer los registros: Adicionar al grupo www-data a la carpeta .ºssec”**usermod -a -G ossec www-data**
- Mostrar la carpeta, **cat /etc/group |grep ossec**
- Dar permisos a la sub carpeta tmp de ossec (usuario,grupos,invitados), **chmod 770 tmp/**
- Ver permisos de la carpeta tmp, **ls -lh**
- Agregar a la carpeta ”tmp.ªl grupo www-data, **chgrp www-data tmp/**
- reiniciar el servicio apache, **systemctl restart apache2**

- reiniciar servicios ossec, **cd /var/ossec/bin** después **./ossec-control restart**
- Ver interfaz gráfica de ossec desde navegador. **localhost/ossec** desde el navegador de preferencia.

E. Anexo: Instalación de Herramienta Mod Security IPS

El primer paso es estar en la consola y entrar como súper usuario con el comando sudo su xxxx, las (xxx) es su clave de acceso, ya al estar como súper usuario se empieza con los comandos de instalación:

- Instalar librería de apache2 **apt-get install libapache2-mod-security2**
- asegurar que está instalado, **apachectl -M | grep security** Debe mostrar security2 module (shared), como se puede ver en la Figura **E-1**

Figura E-1.: Plantilla editada terminada, **Fuente:** Elaboración propia

- renombrar mod a.conf, **mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf**
- renombrar archivo crs, **rm -rf /usr/share/modsecurity-crs**

- descargar reglas crs, **apt-get install git** posteriormente **git clone https://github.com/SpiderL modsecurity-crs.git /usr/share/modsecurity-crs**
- entrar a crs, **cd owasp-modsecurity-crs**
- renombrar el example,**mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf**
- mover reglas a carpeta modsecurity, **mv rules/ /etc/modsecurity/**
- entrar a configurar ruta de reglas, **nano /etc/apache2/mods-available/security2.conf**
- incluir dentro de security2.conf, **Include /etc/modsecurity/rules/*.conf** como se muestra en la Figura **E-2**.

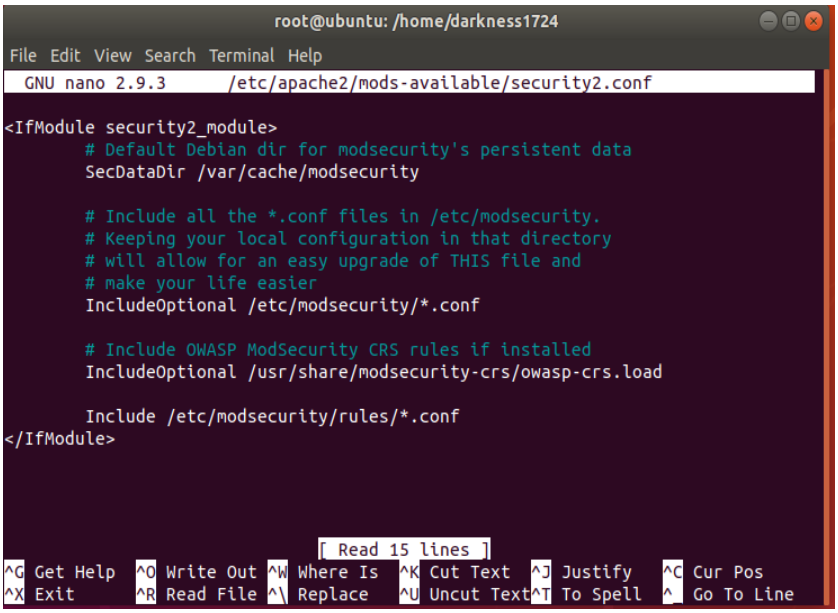
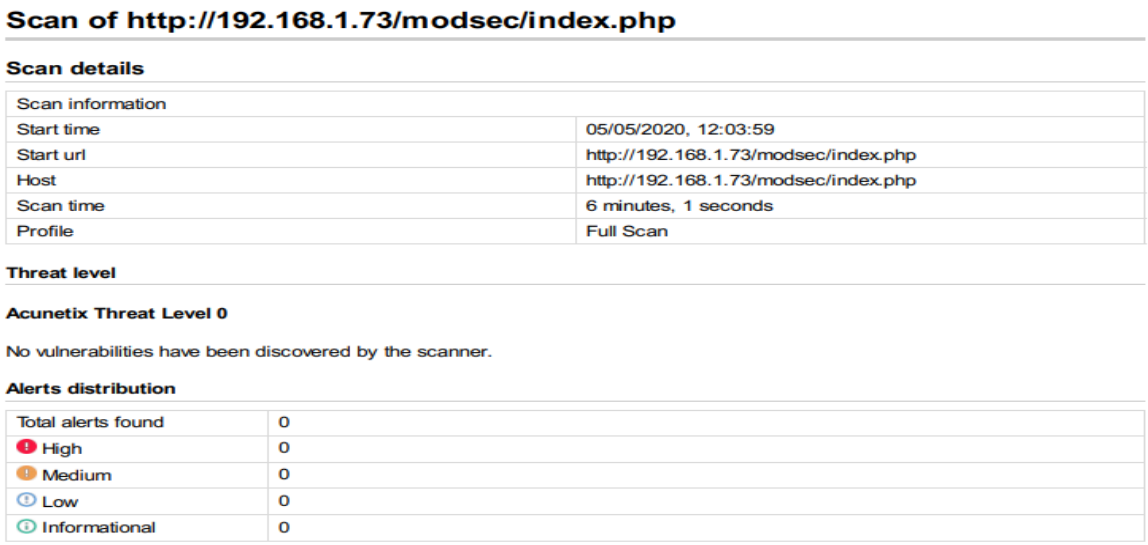


Figura E-2.: Incluir línea de código, **Fuente:** Elaboración propia

- reiniciar apache 2, **service apache2 restart**.
- NO olvidar la ruta de configuración, **/etc/modsecurity/modsecurity.conf**

F. Anexo: Informe Ejecutivo e Iso 27001 de Acunetix

El informe ejecutivo dio por resultado una página la cual se muestra en la Figura **F-1**, pero el informe Iso 27001 dio por resultado un informe extenso por lo tanto, se subió a un servidor para su descarga, el link de su descarga es el siguiente [1].



Scan of http://192.168.1.73/modsec/index.php

Scan details

Scan information	
Start time	05/05/2020, 12:03:59
Start url	http://192.168.1.73/modsec/index.php
Host	http://192.168.1.73/modsec/index.php
Scan time	6 minutes, 1 seconds
Profile	Full Scan

Threat level

Acunetix Threat Level 0

No vulnerabilities have been discovered by the scanner.

Alerts distribution

Total alerts found	0
High	0
Medium	0
Low	0
Informational	0

Figura F-1.: Informe Ejecutivo de Acunetix **Fuente:** Elaboración propia

[1] https://www.mediafire.com/file/i3xsmgddbfr6fvq/ISO27001.pdf/file

G. Anexo: LINK para la Descarga del PROTOTIPO DE FRAMEWORK MODSEC

Cuando se tenga las dos Herramientas instaladas Mod Security y OSSEC se va a la dirección /var/www/html y remplazar la carpeta de OSSEC por la descargada MODSEC, la descarga del Prototipo de Framework se encuentra en el link [1].

Figura G-1.: PROTOTIPO DE FRAMEWORK MODSEC **Fuente:** Elaboración propia

[1] https://www.mediafire.com/file/tkztewm6y9x80pk/modsec.zip/file

H. Anexo: Entrevista a Profesional en el área de sistemas de la entidad Promotora Médica las Américas

Entrevista a la promotora médica Las Américas de la ciudad de Medellín.
Fecha: 07-09-2019
Estudio para conocer el trabajo con el estándar HL7 y su Seguridad.

1. ¿Para qué es Utilizado el estándar HL7 en la promotora Médica Las Américas?
R/ El estándar se lo utiliza para realizar inscripciones de pacientes con datos personales de ellos y ser enviado a una base de datos.

2. ¿Qué versión es utilizado para el estándar HL7?
R/ La versión utilizada es la HL7 V2

3. ¿Sabía usted que existe un estándar de HL7 CDA para Historiales Médicos Electrónicos?
R/ No se tenía conocimiento de un estándar para historiales médicos que trabaje con HL7.

4. ¿Qué seguridad está implementada para la protección de los datos enviados por el estándar HL7?
R/ La seguridad que se tiene para el envío de los datos de los pacientes es la básica para la base de datos, el estándar como tal no posee seguridad.

5. ¿Qué conocimiento tiene usted del estándar HL7?
R/ Sinceramente lo único que se conoce es que sirve para realizar envío de datos de una forma interoperativa.

6. ¿Cree usted que es factible un cambio de estándar para digitalizar los historiales médicos de forma interoperable?
R/ Sí, es factible que las entidades Médicas trabajen todas bajo un estándar interoperable para los historiales, de esa forma poder realizar los envíos sin problemas.

7. ¿Podrían mostrarnos la plataforma con la que trabajan el estándar HL7?
R/ Por políticas de Seguridad de la entidad médica se prohíbe este tipo de información.

8. ¿Podrían Facilitarnos unos datos para ver el almacenamiento de ellos en la base de datos?
R/ como lo dije anteriormente, no es posible por las políticas de Seguridad de la entidad.

9. ¿Cree usted que más adelante la empresa podría cambiar y avanzar en plataforma de envíos?
R/ sería lo más oportuno, pero la verdad no, ya que siguen trabajando con el estándar HL7 en su versión 2 y creo que ya existen muchos más avanzados.

10. ¿cree usted que el envío de datos en HL7 posee buena seguridad?
R/ La verdad no conozco mucho del estándar, así que no se qué tipo de seguridad maneja ni bajo que tipo de seguridad se realiza el envío de estos datos.

Bibliografía

[1] Subcomité Técnico HL7 V3-CDA. Guía para el desarrollo de documentos CDA. pages 1–60, 2007.

[2] Jairo Alejandro Buitrago, Andrea Torres, and Rosmary Martínez-rueda. con actividad física en población adulta y adulto mayor de zonas rurales en Colombia. (1):2018, 2018.

[3] Kwangsoo Seol, Young-Gab Kim, Euijong Lee, Young-Duk Seo, and Doo-Kwon Baik. Privacy-preserving attribute-based access control model for xml-based electronic health record system. *IEEE Access*, 6:9114–9128, 2018.

[4] Esarrollo Del and Eneral De. Arquitectura de referencia para el intercambio de información entre sistemas de información de registro electrónico para la salud. 0.4:1–24, 2012.

[5] M. I. Sabar, Prasad M. Jayaweera, and E. A.T.A. Edirisuriya. International Interoperability through unified universal HL7 v3 Green Messaging. *15th International Conference on Advances in ICT for Emerging Regions, ICTer 2015 - Conference Proceedings*, pages 112–118, 2016.

[6] Selene Indarte and Pablo Pazos Gutiérrez. Estándares e interoperabilidad en salud electrónica: Requisitos para una gestión sanitaria efectiva y eficiente. Documentos de Proyectos 440, Naciones Unidas Comisión Económica para América Latina y el Caribe (CEPAL), October 2011.

[7] Oscar Bernal-Acevedo and Juan Camilo Forero-Camacho. Sistemas de información en el sector salud en colombia. *Revista Gerencia y Políticas de Salud*, 10(21):85–100, 2011.

[8] Laura Martinez, Andres Soto, Luis Eraso, Armando Ord, and Hugo Ordo. *Advances in Computing and Data Sciences*, volume 721 of *Communications in Computer and Information Science 721*. Springer Singapore, 1 edition, 2017.

[9] Enrique Mario, Ing Cortés. 4to Foro El sistema de salud en Colombia. *Universidad del Cauca*, pages 1–8, 2011.

[10] Manuel Leos Rivas. InfoSec Reading Room Attack and Defend Linux Privilege Escalation The Institute , A hoeta insll Rights. *Securing the Home IoT Network*, (Security 401):1–32, 2017.

[11] Miguel Pedrera Jiménez. Servicio web para el acceso a información sanitaria de pacientes renales. 2016.

[12] O OMS. *Revisión de Estándares de Interoperabilidad para la eSalud en Latinoamérica y el Caribe*. 2016.

[13] Calero Suntasig and Henry Daniel. Análisis de vulnerabilidades para la infraestructura de red de la bolsa de valores de quito, aplicando una metodología de ethical hacking. 2020.

[14] Andrew Hay, Daniel Cid, and Rory Bray. *OSSEC Host-Based Intrusion Detection Guide*. Syngress Publishing, 2008.

[15] Manuel Arostegui. Securizando apache con modsecurity. *Todo linux: la revista mensual para entusiastas de GNU/LINUX*, (82):30–34, 2007.

[16] Top OWASP. Top 10-2017. *The Ten Most Critical Web Application Security Risks. OWASP™ Foundation. The free and open software security community. URL: https://www. owasp. org/index. php/Top_10-2017_Top_10*, 2017.

[17] John L Burns, Dan Hasting, Judy W Gichoya, Ben McKibben, Lindsey Shea, and Mark Frank. Just in time radiology decision support using real-time data feeds. *Journal of Digital Imaging*, 33(1):137–142, 2020.

[18] Carlos Andrew Costa Bezerra, André Magno Costa de Araújo, and Valéria Cesário Times. An hl7-based middleware for exchanging data and enabling interoperability in healthcare applications. In *17th International Conference on Information Technology–New Generations (ITNG 2020)*, pages 461–467. Springer, 2020.

[19] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[20] Marco Eichelberg, Klaus Kleber, and Marc Kämmerer. Cybersecurity challenges for pacs and medical imaging. *Academic Radiology*, 2020.

[21] BB Gupta and Pooja Chaudhary. *Cross-site Scripting Attacks: Classification, Attack, and Countermeasures*. CRC Press, 2020.

[22] Víctor Chavarria Gonzalez. Estudio de los ataques contra website. owasp. 2020.

[23] Ashara Banu Mohamed, Norbik Bashah Idris, and Bharanidharan Shanmugum. A brief introduction to intrusion detection system. In S. G. Ponnambalam, Jussi Parkkinen, and Kuppan Chetty Ramanathan, editors, *Trends in Intelligent Robotics, Automation, and Manufacturing*, pages 263–271, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[24] Andrew Kronser et al. Common vulnerabilities and exposures: Analyzing the development of computer security threats. 2020.

[25] Huber Espinoza-Palma. Framework de seguridad dinámica consultores. url http://hdl.handle.net/2238/2781, 2011. Accedido 06-08-2020.

[26] Robert H Dolin, Gay Giannone, and Gunther Schadow. Enabling joint commission medication reconciliation objectives with the hl7/astm continuity of care document standard. In *AMIA Annual Symposium Proceedings*, volume 2007, page 186. American Medical Informatics Association, 2007.

[27] Shahid Munir Shah and Rizwan Ahmed Khan. Secondary use of electronic health record: Opportunities and challenges. *arXiv preprint arXiv:2001.09479*, 2020.

[28] Reference Dallas Haselhorst. HL7 Data Interfaces in Medical Environments: Understanding the Fundamental Flaw in Healthcare. *SANS*, 2017.

[29] Suranga N Kasthurirathne, Burke Mamlin, Harsha Kumara, Grahame Grieve, and Paul Biondich. Enabling Better Interoperability for HealthCare: Lessons in Developing a Standards Based Application Programing Interface for Electronic Medical Record Systems. *Journal of Medical Systems*, 39(11):182, oct 2015.

[30] Combining Ontologies and Open Standards to Derive a Middle Layer Information Model for Interoperability of Personal and Electronic Health Records. *Journal of Medical Systems*, 41(12):195, oct 2017.

[31] A Security Framework for Nationwide Health Information Exchange based on Telehealth Strategy. *Journal of Medical Systems*, 39(5):51, mar 2015.

[32] Humberto F Mandirola, Ing Cesar Moreno, Mandirola H Brieux F, La F Rosa, and Moreno C Ricardo Herrero -Lic Jorge A Guerra. Integrando aplicaciones médicas con HL7 lecciones aprendidas CURSO VIRTUAL SOBRE MIRTH CONNECT Introducción a los motores de interoperabilidad con estándares. HL7 LATAM NEWS Seguridad de la Información en entornos de salud. 2017.

[33] Yanssel Urquijo Morales and C Arturo Orellana García. Un enfoque actual para garantizar la seguridad de la información del sistema xavia his aplicando la pki nacional a current approach to ensuring the security of the system's information xavia his applying the national. 2020.

[34] P Moreno and G Bastidas. Propuesta metodológica de intercambio electrónico de información clínica basada en estándares de telemedicina/methodological proposal of electronic data interchange and clinical information based on telemedicine standards. *KnE Engineering*, pages 206–231, 2020.

[35] Nahuel Hourcade, Valentina Coggan, and Estefanía Della Mea. Memoria de un proyecto de informática médica presentada a la facultad de ingeniería. 2020.

[36] Edgar Lugo, Angel Villegas, Hyxia Villegas, and Jenny Pacheco. Diseño de un software para la interpretación de historias clínicas electrónicas basadas en hl7/cda aplicado en servicios de telemedicina. *Revista INGENIERÍA UC*, 15:31–40, 01 2008.

[37] Joshua Mandel. Security vulnerabilities in c-cda display using cda.xsl. url https://smarthealthit.org/2014/04/security-vulnerabilities-in-ccda-display/, 2014. Accedido 03-01-2020.

[38] The Backbeach Software C-CDA Viewer. C-cda® rendering tool challenge. url http://backbeachsoftware.com.au/?cmd=getsqlsid=id$_s ite = 1 id_t ag = 52, 2016. Accedido 05 - 03 - 2020.$

[39] Md Iftekhar Hussain. Internet of things: challenges and research opportunities. *CSI transactions on ICT*, 5(1):87–95, 2017.

[40] HyunHo Kim, HoonJae Lee, and HyoTaek Lim. Performance of packet analysis between observer and wireshark. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pages 268–271. IEEE, 2020.

[41] John D D'Amore, Joshua C Mandel, David A Kreda, Ashley Swain, George A Koromia, Sumesh Sundareswaran, Liora Alschuler, Robert H Dolin, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. Are Meaningful Use Stage 2 certified EHRs ready for interoperability? Findings from the SMART C-CDA Collaborative. *Journal of the American Medical Informatics Association*, 21(6):1060–1068, 06 2014.