# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA DE SISTEMAS

## SISTEMA DE RECOMENDACIÓN PARA CIBERSEGURIDAD BASADO EN UN ENFOQUE DE RATINGS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**CARLOS ANTONIO AYALA TIPÁN**
carlos.ayala01@epn.edu.ec

**KEVIN ORLANDO JIMENEZ SARAGURO**
kevin.jimenez@epn.edu.ec

**DIRECTOR: PhD. EDISON FERNANDO LOZA AGUIRRE**
edison.loza@epn.edu.ec

**CODIRECTOR: MSc. ROBERTO OMAR ANDRADE PAREDES**
roberto.andrade@epn.edu.ec

**Quito, abril 2020**

# CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **Carlos Antonio Ayala Tipán** y **Kevin Orlando Jimenez Saraguro**, bajo mi supervisión.

**PhD. Edison Fernando Loza Aguirre**

**DIRECTOR DE PROYECTO**

# CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **Carlos Antonio Ayala Tipán** y **Kevin Orlando Jimenez Saraguro**, bajo mi supervisión.

**MSc. Roberto Omar Andrade Paredes**

**CODIRECTOR DE PROYECTO**

# DECLARACIÓN

Nosotros, **Carlos Antonio Ayala Tipán** y **Kevin Orlando Jimenez Saraguro**, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

**Carlos Antonio Ayala Tipán**

**Kevin Orlando Jimenez Saraguro**

# DEDICATORIA

A mis padres, Marcelino y Narciza, a mi hermana Michelle por su cariño, paciencia y ánimo al impulsarme a superarme cada día, fueron los primeros en creer en mí y brindarme su apoyo incondicional durante toda mi vida académica.

**Carlos A. Ayala T.**

# DEDICATORIA

A mis padres, Marlene y Orlando, a mi hermana Katherin, por ser testigos de mi esfuerzo y dedicación durante toda la mi vida académica, por su gran apoyo y amor hasta el final.

A toda mi familia quienes nunca dejaron de creer en mí.

**Kevin O. Jimenez S.**

# AGRADECIMIENTO

En primer lugar, quiero expresar mi total agradecimiento al PhD. Edison Loza y MSc. Roberto Andrade director y codirector de este proyecto de investigación, por todo el apoyo brindado y la amabilidad en todo el proceso de la realización de este trabajo, dándonos una excelente apertura a nuestras ideas y brindando una dirección a las mismas.

Asimismo, agradezco a mi compañero y amigo Kevin Jimenez el cual colaboró y ha sido una parte esencial dentro de la realización de este trabajo, y con quien he tenido la dicha de compartir proyectos dentro de mi formación.

Agradezco a mis padres, que me criaron con valores los cuales formaron a la persona en la que soy ahora, son los promotores de mis anhelos, gracias por la paciencia y comprensión. A mi hermana, por su cariño y amor que sin importar de peleas somos hermanos y siempre estaremos juntos. A ustedes, que confiaron y nunca han dudado de mí en todos los aspectos de mi vida, les agradezco por su amor y tiempo que me han concedido. ¡Cada logro ha sido y será por y para ustedes!

A mis amigos, con quienes compartí maravillosos momentos dentro y fuera del aula, por su cariño y amistad de manera desinteresada. Y, a todas aquellas personas que aportaron con un granito de arena en la obtención de un logro más en mi visa.

Por último y no menos importante, a cada uno de mis maestros que ayudaron en el enriquecimiento de todo el conocimiento y habilidades obtenidas a lo largo de estos años dentro de la Escuela Politécnica Nacional.

**Carlos A. Ayala T.**

# AGRADECIMIENTO

Agradezco a Dios, por haberme ayudado a culminar esta gran etapa de mi vida, por hacer que este sueño se haya cumplido.

A mi madre Marlene, mi padre Orlando, los cuales me brindaron su gran apoyo incondicional, durante toda esta travesía, y su gran amor. Fueron mi gran inspiración para seguir adelante y no darme por vencido. A mi madre, por estar siempre pendiente de mí. A mi padre, por brindarme sus sabias consejos en momentos difíciles. A mi hermana, Katherin, por estar siempre a mi lado en las buenas y en las malas. A todos ellos les agradezco porque nunca dejaron de creer en mí.

A mis tíos, primos, abuelitos, quienes nunca me dieron la espalda, fueron un gran pillar en toda mi vida académica, por su gran apoyo moral y amor.

A mi director PhD. Edison Loza y codirector MSc. Roberto Andrade, por su gran apoyo durante todo el desarrollo de este proyecto, por sus sabios consejos, sabiduría y tiempo.

Agradezco a mi compañero Carlos Ayala, por su gran amistad, colaboración y gran apoyo en la realización del presente trabajo, por su tiempo compartido y conocimientos.

Sin olvidarme de la persona que estuvo en esta última etapa de mi vida, que ahora forma parte de mí y de mi vida, y con quien he tenido la dicha de compartir gratos momentos, gracias por soportarme en las buenas y en las malas, agradezco su gran apoyo y amor brindado.

A mis amigos, los cuales compartí gratos momentos en toda mi vida académica, que ahora son parte de mi vida. Les agradezco su gran apoyo y amistad.

Finalmente, agradezco a la Escuela Politécnica Nacional y a sus autoridades por haberme brindado la oportunidad de haber culminado mis estudios y una etapa más de mi vida.

**Kevin O. Jimenez S.**

# ÍNDICE DE CONTENIDO

**ÍNDICE DE FIGURAS** ..... **XII**

**ÍNDICE DE TABLAS** ..... **XIV**

**RESUMEN** ..... **XVI**

**ABSTRACT** ..... **XVII**

**1. INTRODUCCIÓN** ..... **1**

1.1. ANTECEDENTES ..... 1

1.2. OBJETIVOS ..... 3

*1.2.1. Objetivo General* ..... *3*

*1.2.2. Objetivos Específicos* ..... *3*

*1.2.3. Alcance* ..... *3*

1.3. MARCO TEÓRICO ..... 3

*1.3.1. Ciberseguridad* ..... *3*

*1.3.2. Tipos de incidentes de ciberseguridad* ..... *4*

*1.3.3. Sistemas de recomendación* ..... *5*

*1.3.4. Tipos de sistemas de recomendación* ..... *5*

1.3.4.1. Sistemas de recomendación basados en filtrado colaborativo ..... 5

1.3.4.2. Sistemas de recomendación basados en contenido ..... 6

1.3.4.3. Sistemas de recomendación basados en conocimiento ..... 6

1.3.4.4. Sistemas de recomendación híbridos ..... 6

1.3.4.5. Sistemas de recomendación demográfica ..... 6

1.3.4.6. Sistemas de recomendación basado en palabras clave ..... 6

*1.3.5. Ventajas y desventajas de los sistemas de recomendación* ..... *7*

1.4. HERRAMIENTAS DE DESARROLLO ..... 8

**2. METODOLOGÍA** ..... **9**

2.1. DESCRIPCIÓN DE SCRUM ..... 9

2.2. PROYECTO SCRUM ..... 12

*2.2.1. Investigación Preliminar* ..... *12*

*2.2.2. Roles* ... *13*
*2.2.3. Requerimientos* ... *13*
*2.2.4. Product Backlog* ... *14*
*2.2.5. Release planning* ... *14*
*2.2.6. Sprint 0* ... *15*
2.2.6.1. Sprint Planning ... 15
2.2.6.2. Ejecución del Sprint ... 15
2.2.6.3. Inspección y adaptación ... 15
*2.2.7. Sprint 1* ... *16*
2.2.7.1. Sprint Planning ... 16
2.2.7.2. Ejecución del Sprint ... 16
2.2.7.3. Inspección y adaptación ... 17
*2.2.8. Sprint 2* ... *17*
2.2.8.1. Sprint Planning ... 18
2.2.8.2. Ejecución del Sprint ... 18
2.2.8.3. Inspección y adaptación ... 18
*2.2.9. Sprint 3* ... *19*
2.2.9.1. Sprint Planning ... 19
2.2.9.2. Ejecución del Sprint ... 19
2.2.9.3. Inspección y adaptación ... 20
*2.2.10. Sprint 4* ... *20*
2.2.10.1. Sprint Planning ... 20
2.2.10.2. Ejecución del Sprint ... 20
2.2.10.3. Inspección y adaptación ... 21
*2.2.11. Sprint 5* ... *21*
2.2.11.1. Sprint Planning ... 22
2.2.11.2. Ejecución del Sprint ... 22
2.2.11.3. Inspección y adaptación ... 22

*2.2.12. Sprint 6* ... *23*
2.2.12.1. Sprint Planning ... 23
2.2.12.2. Ejecución del Sprint ... 23
2.2.12.3. Inspección y adaptación ... 24
*2.2.13. Sprint 7* ... *24*
2.2.13.1. Sprint Planning ... 24
2.2.13.2. Ejecución del Sprint ... 24
2.2.13.3. Inspección y adaptación ... 25
2.3. EVALUACIÓN ... 25
*2.3.1. Pruebas de caja negra* ... *25*
**3. RESULTADOS Y DISCUSIÓN ... 27**
3.1. DESCRIPCIÓN DEL SISTEMA ... 27
*3.1.1. Arquitectura del Sistema de recomendación* ... *27*
3.1.1.1. Arquitectura Lógica ... 27
3.1.1.2. Arquitectura Física ... 28
*3.1.2. Descripción de los artefactos* ... *29*
3.1.2.1. Sistema de Recomendación ... 29
3.2. RESULTADOS DE LA EVALUACIÓN ... 31
*3.2.1. Resultados Prueba de caja negra* ... *31*
*3.2.2. Resultados Encuesta a expertos* ... *31*
3.3. DISCUSIÓN ... 35
**4. CONCLUSIONES Y RECOMENDACIONES ... 37**
4.1. CONCLUSIONES ... 37
4.2. RECOMENDACIONES ... 38
**ANEXO ... I**
**ANEXO I. ARTÍCULO CIENTÍFICO ... I**
I.1 NOTIFICACIÓN DE RECEPCIÓN POR PARTE DE SAM'20 ... I
I.2 ARTÍCULO ENVIADO A SAM'20 ... II

**ANEXO II. TABLAS RELACIONADAS AL PROCESO DE SCRUM ..... X**

II.1 EJECUCIÓN DE FASES SPRINT 0 ..... X

II.2 EJECUCIÓN DE FASES SPRINT 1 ..... X

II.3 EJECUCIÓN DE FASES SPRINT 2 ..... XI

II.4 EJECUCIÓN DE FASES SPRINT 3 ..... XII

II.5 EJECUCIÓN DE FASES SPRINT 4 ..... XII

II.6 EJECUCIÓN DE FASES SPRINT 5 ..... XIII

II.7 EJECUCIÓN DE FASES SPRINT 6 ..... XIV

II.8 EJECUCIÓN DE FASES SPRINT 7 ..... XV

# ÍNDICE DE FIGURAS

**Figura 1.1:** Ciberespacio ..... 4

**Figura 2.1:** Proceso de SCRUM..... 9

**Figura 2.2:** Ciclo de vida de SCRUM ..... 10

**Figura 2.3:** Ciclo de Sprint en Scrum ..... 11

**Figura 2.4:** Ejecución del Sprint 0 ..... 15

**Figura 2.5:** Ejecución del Sprint 1 ..... 17

**Figura 2.6:** Ejecución del Sprint 2 ..... 18

**Figura 2.7:** Ejecución del Sprint 3 ..... 19

**Figura 2.8:** Ejecución del Sprint 4 ..... 21

**Figura 2.9:** Ejecución del Sprint 5 ..... 22

**Figura 2.10:** Ejecución del Sprint 6 ..... 23

**Figura 2.11:** Ejecución del Sprint 7 ..... 25

**Figura 2.12:** Contexto de funcionalidad de pruebas de caja negra..... 26

**Figura 2.13:** Prueba de caja negra..... 26

**Figura 3.1:** Estructura de la información ..... 27

**Figura 3.2:** Arquitectura Lógica del sistema..... 28

**Figura 3.3:** Arquitectura Física del sistema ..... 28

**Figura 3.4:** Ingreso de la anomalía ..... 29

**Figura 3.5:** Ingreso de la anomalía ..... 29

**Figura 3.6:** Mensaje informativo anomalía no registrada ..... 30

**Figura 3.7:** Autocomplete de la anomalía a buscar ..... 30

**Figura 3.8:** Los cinco ataques similares..... 30

**Figura 3.9:** Descripción y recomendación para la anomalía buscada ..... 30

**Figura 3.10:** Modelo de encuesta..... 32

**Figura 3.11:** Resultados de tiempo promedio de respuesta efectiva ..... 33

**Figura 3.12:** Resultados de frecuencia de presencia de anomalías ..... 33

**Figura. I.1.** Notificación de recepción a carlos.ayala01@epn.edu.ec ..... I

**Figura. I.2.** Notificación de recepción a kevin.jimenez@epn.edu.ec..... I

**Figura. I.3.** Página carátula de artículo científico..... II

**Figura. I.4.** Página uno de artículo científico..... III

**Figura. I.5.** Página dos de artículo científico..... IV

**Figura. I.6.** Página tres de artículo científico..... V

**Figura. I.7.** Página cuatro de artículo científico..... VI

**Figura. I.8.** Página cinco de artículo científico ..... VII

**Figura. I.9.** Página seis de artículo científico ..... VIII

**Figura. I.10.** Página siete de artículo científico ..... IX

# ÍNDICE DE TABLAS

**Tabla 1.1:** Tipos de incidentes ..... 4

**Tabla 1.2:** Ventajas y desventajas de los sistemas de recomendación ..... 7

**Tabla 1.3:** Herramientas utilizadas para el desarrollo ..... 8

**Tabla 2.1:** Etapas Sprint ..... 12

**Tabla 2.2:** Roles SCRUM ..... 13

**Tabla 2.3:** Lista de Historias de Usuarios ..... 14

**Tabla 2.4:** Planificación Sprint ..... 15

**Tabla 3.1:** Prueba de caja negra 01: Ingreso de la anomalía a ser mitigada ..... 31

**Tabla 3.2:** Prueba de caja negra 02: Visualizar anomalías similares ..... 31

**Tabla 3.3:** Prueba de caja negra 03: Visualizar la recomendación y su descripción ..... 31

**Tabla II.1:** Fechas de las fases del Sprint 0 ..... X

**Tabla II.2:** Backlog Sprint 0 ..... X

**Tabla II.3:** Criterios de aceptación Sprint 0 ..... X

**Tabla II.4:** Fechas de las fases del Sprint 1 ..... X

**Tabla II.5:** Backlog Sprint 1 ..... XI

**Tabla II.6:** Criterios de aceptación Sprint 1 ..... XI

**Tabla II.7:** Fechas de las fases del Sprint 2 ..... XI

**Tabla II.8:** Backlog Sprint 2 ..... XI

**Tabla II.9:** Criterios de aceptación Sprint 2 ..... XI

**Tabla II.10:** Fechas de las fases del Sprint 3 ..... XII

**Tabla II.11:** Backlog Sprint 3 ..... XII

**Tabla II.12:** Criterios de aceptación Sprint 3 ..... XII

**Tabla II.13:** Fechas de las fases del Sprint 4 ..... XII

**Tabla II.14:** Backlog Sprint 4 ..... XIII

**Tabla II.15:** Criterios de aceptación Sprint 4 ..... XIII

**Tabla II.16:** Fechas de las fases del Sprint 5 ..... XIII

**Tabla II.17:** Backlog Sprint 5 ..... XIII

**Tabla II.18:** Criterios de aceptación Sprint 5 .....XIV

**Tabla II.19:** Fechas de las fases del Sprint 6 .....XIV

**Tabla II.20:** Backlog Sprint 6.....XIV

**Tabla II.21:** Criterios de aceptación Sprint 6 .....XIV

**Tabla II.22:** Fechas de las fases del Sprint 7 .....XV

**Tabla II.23:** Backlog Sprint 7.....XV

**Tabla II.24:** Criterios de aceptación Sprint 8 .....XV

# RESUMEN

La principal función de un analista de seguridad informática es proteger y tomar las mejores decisiones para salvaguardar la integridad de las redes y sistemas informáticos dentro de su organización. En sus actividades diarias, el analista tiende a encontrarse con un sin número de riesgos, anomalías y vulnerabilidades que comprometen lo anteriormente expuesto. Para poder brindar una pronta respuesta, el analista depende en general de su buen juicio al momento de tomar decisiones. Al enfrentarse a un sin número de riesgos, anomalías y vulnerabilidades diarias, los analistas están sujetos a desbordamientos, dando respuesta a solo unas pocas de estas amenazas, o simplemente, dando respuesta a los que tienen mayor nivel de criticidad.

En este sentido, en esta investigación nos enfocamos de brindar una herramienta la cual ayude y facilite al analista de seguridad a filtrar las anomalías, vulnerabilidades y riesgos latentes, los cuales se registren dentro de esta herramienta. Para cumplir este objetivo, se desarrolló un sistema de recomendación basado en filtrado colaborativo y en el conocimiento, generando ratings de los peores casos con las mejores recomendaciones ayudados de un juicio de expertos.

Este sistema de recomendación se evaluó con criterios de aceptación de tecnología en pruebas de caja negra, tomando en cuenta la utilidad como la facilidad de uso, que presenta el sistema. Con dichos criterios se obtuvo resultados donde se evidenció una mejora en el tiempo de respuesta ante los problemas. Además, se eliminó la subjetividad del analista y se redujo el número de procesos manuales.

**Palabras clave:** sistema de recomendación, filtrado colaborativo, conocimiento, analista de seguridad, base de conocimiento.

# ABSTRACT

The primary role of a computer security analyst is to protect and make the best decisions to safeguard the integrity of computer networks and systems. In his daily activities, the analyst tends to encounter several risks, anomalies and vulnerabilities that compromise security. In order to provide a prompt response, the analyst generally depends on good judgment when making decisions. Faced with countless daily risks, anomalies and vulnerabilities, analysts are subject to spillovers, responding to only a few of these threats, or simply responding to those with the highest level of criticality.

Thus, in this research we focus on providing a tool to help security analysts to filter anomalies, vulnerabilities and latent risks. To meet this objective, a recommendation system based on collaborative filtering and knowledge was developed, generating ratings of the worst cases with the best recommendations aided by expert judgment.

The recommendation system was evaluated with technology acceptance criteria in black box tests, considering the usefulness and ease of use of the system. With these criteria, results were obtained where an improvement in response time to problems was evident. Furthermore, the analyst's subjectivity was eliminated, and the number of manual processes was reduced.

**Keywords:** recommendation system, collaborative filtering, knowledge, security analyst, knowledge base.

# 1. INTRODUCCIÓN

## 1.1. ANTECEDENTES

Un analista de ciberseguridad según, Randall Fietzsche [1], es quien analizará el riesgo y analizará la amenaza que pueda comprometer su organización. Los analistas de ciberseguridad planifican y llevan a cabo medidas de seguridad con el objetivo de proteger las redes y los sistemas informáticos de la organización. En otras palabras, su trabajo es ayudar a la organización a comprender lo que está sucediendo y hacia donde debe ir en términos de seguridad informática.

Según la Oficina de Estadísticas Laborales de EE. UU. (BLS), los deberes que se enfrentan analistas de ciberseguridad son [2]:

- Supervisar las redes de su organización en busca de violaciones de seguridad e investigar una violación cuando ocurra.
- Instalar y usar software, como firewalls y programas de cifrado de datos, para proteger la información confidencial.
- Preparar informes que documenten violaciones de seguridad y el alcance de los daños causados por las violaciones.
- Realizar pruebas de penetración, que es cuando los analistas simulan ataques para buscar vulnerabilidades en sus sistemas antes de que puedan ser explotados.
- Investigar las últimas tendencias en seguridad de tecnología de la información (TI).
- Desarrollar estándares de seguridad y mejores prácticas para su organización.
- Recomendar mejoras de seguridad al personal directivo o de alto nivel de TI.
- Ayudar a los usuarios de computadoras cuando necesiten instalar o conocer nuevos productos y procedimientos de seguridad.

El trabajo de estos analistas consiste en día a día encontrarse con riesgos, vulnerabilidades y amenazas dentro de las organizaciones, llevándolos a buscar un framework de trabajo específico y de esta manera priorizar los principales incidentes y ataques con el fin de obtener las mejores acciones para contrarrestarlos. Sin embargo, existen tres factores que pueden llegar a afectar sus decisiones [3]:

1. Tiempo.

2. Procesos y metodologías manuales.
3. Subjetividad de analista.

El desafío que se le presenta a cualquier analista de ciberseguridad sea cual sea el entorno en el que se desempeñe, es el de poder detectar los diferentes ataques cibernéticos. Sobre todo, cuando la forma tradicional de detección tiende a reportar grandes cantidades de alertas, las cuales deben ser examinadas y verificadas con toda la información, estructurada y no estructurada, disponible [4].

Los desafíos presentes para un analista crecen a medida que transcurre el tiempo ya que día a día surgen gran variedad de ataques cibernéticos; los cuales son cada vez más sofisticados y presentan diversidad de formas. Estos ataques irán variando en función del comportamiento y personalidad del atacante; dificultando las tareas de los analistas [5].

Todos estos desafíos hacen que un analista se sienta desbordado al intentar discriminar que producto, contenido o servicio cubre las necesidades correctas para resolver un problema de forma óptima [6]. Para ayudarlos en su búsqueda se pueden utilizar sistemas de recomendación que “producen recomendaciones personalizadas como salida o tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles de entre una gran cantidad de productos disponibles” [7]; en este caso refiriéndonos al producto como toda la información existente.

Para el usuario, un sistema de recomendación hace que la búsqueda de productos/información interesante sea más sencilla, reduciendo el número de opciones que tiene que revisar y mejorando su experiencia [8]; brindándole así alternativas para una adecuada toma de decisiones partiendo de una base sólida de conocimiento, lo que conlleva a alivianar sus tareas.

El sistema de recomendación fue evaluado por un grupo de expertos en el área de seguridad informática y análisis de datos que llevaron a cabo su evaluación utilizando el marco del Modelo de Aceptación de Tecnología (TAM) [9]. Por lo tanto, la evaluación se basó en los dos pilares principales de TAM: (1) la utilidad percibida definida como la probabilidad subjetiva de una persona que, al usar un determinado sistema, mejoraría su desempeño en el trabajo; y (2) la facilidad de uso percibida que se refiere al grado en que una persona cree que usar cierto sistema será fácil [9].

El presente trabajo ha servido como base para la realización del artículo científico titulado “A hybrid recommender system for Cybersecurity based on a rating approach”, mismo que se ha sido enviado a la conferencia: The 2020 International Conference on

Security and Management (SAM'20) http://sam.udmercy.edu/sam20/index.html y se encuentra dentro del Anexo I.2.

## 1.2. OBJETIVOS

### 1.2.1. Objetivo General

Elaborar un sistema de recomendación para ciberseguridad basado en un enfoque de ratings.

### 1.2.2. Objetivos Específicos

- Analizar las recomendaciones colaborativas, basado en contenido y conocimiento.
- Elaborar ratings con un juicio de expertos.
- Priorizar los principales riesgos, vulnerabilidades, amenazas y las mejores recomendaciones a los incidentes y ataques, las cuales un analista de ciberseguridad puede presentar en el día a día dentro de su organización

### 1.2.3. Alcance

En el presente proyecto se propone el desarrollo de un sistema de recomendación para alivianar las tareas y problemas que puede presentar un analista de ciberseguridad. El sistema priorizará vulnerabilidades, amenazas y riesgos, y las posibles soluciones a las mismas. De esta manera se pretende mejorar el tiempo de respuesta antes los diferentes incidentes por parte del analista. Este sistema proporcionará las mejores respuestas para cada incidente de seguridad basado en ratings clasificados con la ayuda de expertos en el área.

## 1.3. MARCO TEÓRICO

### 1.3.1. Ciberseguridad

La ciberseguridad nació cuando apareció el ciberespacio. El ciberespacio está conformado por la red de computadores y los usuarios o personas, el mundo electrónico creado por redes interconectadas de tecnología de la información y la información en esas redes. El ciberespacio ayuda a conectar a las personas, permitiéndoles intercambiar ideas, brindar servicios, etc. El ciberespacio sigue evolucionando conforme pasa el tiempo, esto significa que no es estático más bien es dinámico, y al igual que el ciberespacio evoluciona, los ciberataques también lo hacen. Esto demanda tomar medidas de precaución, ahí es donde interviene la ciberseguridad como mecanismo

para poder mitigar los ataques (Figura 1.1) [10] [11].

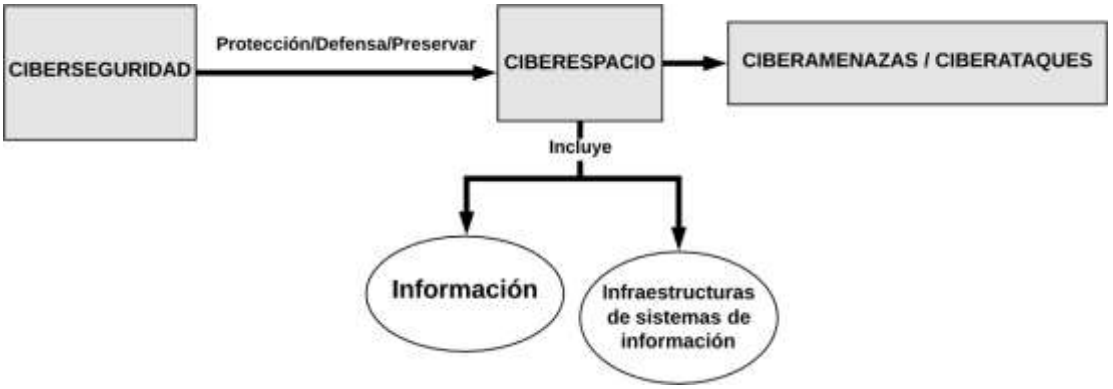



**Figura 1.1:** Ciberespacio [11]. Elaborado por autores.

La ciberseguridad es la forma de proteger información de cualquier ataque digital, ya que el objetivo de los ataques digitales es poder acceder, modificar o eliminar información. Esto es particularmente desafiante hoy porque hay más dispositivos que personas, y los atacantes se están volviendo más innovadores [12]. La ciberseguridad se enfoca también en brindar métodos de defensa para poder detectar y capturar a cualquier intruso que desea ingresar a cualquier sistema de información [13].

### 1.3.2. Tipos de incidentes de ciberseguridad

Un incidente en ciberseguridad es un evento o un conjunto de eventos no deseados o inesperados, el cual impacta negativamente a los procesos y operaciones de las organizaciones o de los negocios. Su objetivo es inhabilitar el uso de la información, eliminarla o modificarla mediante la corrupción de los sistemas de información a través de infecciones por malware, phishing, etc [14].

Los incidentes de ciberseguridad se clasifican en los siguientes:

**Tabla 1.1:** Tipos de incidentes [12]. Elaborado por autores.

| Incidente | Descripción |
|---|---|
| **Phishing** | El phishing es la práctica de enviar correos electrónicos fraudulentos que se asemejan a correos electrónicos de fuentes confiables. El objetivo es robar datos confidenciales como números de tarjetas de crédito e información de inicio de sesión. Es el tipo más común de ciberataque. Puede ayudar a protegerse a través de la educación o una solución tecnológica que filtra los correos electrónicos maliciosos. |
| **Ransomware** | El ransomware es un tipo de software malicioso. Está diseñado para extorsionar dinero bloqueando el acceso a los archivos o al sistema informático hasta que se pague el rescate. Pagar el rescate no garantiza que los archivos se recuperarán o que se restaurará el sistema. |
| **Malware** | El malware es un tipo de software diseñado para obtener acceso no autorizado o para dañar una computadora. |
| **Ingeniería social** | La ingeniería social es una táctica que usan los adversarios para engañarte y revelar información confidencial. Pueden solicitar un pago monetario u obtener acceso a sus datos confidenciales. La ingeniería social se puede combinar con cualquiera de las amenazas enumeradas anteriormente para que sea más probable que haga clic en enlaces, |

| | |
|---|---|
| | descargue malware o confíe en una fuente maliciosa. |

### 1.3.3. Sistemas de recomendación

Los sistemas de recomendación son técnicas y herramientas de software que proporcionan como sugerencias un subconjunto de elementos, pertenecientes a un universo de alternativas, que se consideran los más apropiados para el usuario. El sistema de recomendación se define como una estrategia de toma de decisiones [15] [16].

Así, uno de los pilares fundamentales de los sistemas de información es la gran cantidad de información que pueden manejar, para poder entregar una recomendación valida. Para brindar estas recomendaciones se toma en cuenta el conocimiento o la experiencia del usuario. Si el usuario no posee conocimiento o experiencia, se podrá brindar una recomendación validad en la categoría que el usuario haya seleccionado [17].

La mayoría de los sistemas de recomendación está enfocado a los usuarios, emitiendo recomendaciones basadas en la similitud de búsquedas que hayan tenido con otros usuarios, o mediante la calificación un usuario que haya dado a un elemento opción.

### 1.3.4. Tipos de sistemas de recomendación

Existen diferentes tipos de sistemas de recomendación, cada uno de ellos tiene su propio enfoque y brinda su propia recomendación basado en su enfoque. Así, a estos sistemas se los puede clasificar en [17] [18]:

- Los sistemas de recomendación basados en filtrado colaborativo.
- Los sistemas de recomendación basados en contenido.
- Los sistemas de recomendación basados en conocimiento.
- Los sistemas de recomendación híbridos.
- Los sistemas de recomendación demográfica.
- Los sistemas de recomendación basado en palabras clave.

#### 1.3.4.1. Sistemas de recomendación basados en filtrado colaborativo

Este tipo de sistema recomendación está enfocado en los ítems en los cuales han recibido una calificación brindada por los usuarios a quienes se les entrego ese ítem u opción. Este tipo de sistema es el más usado ya que ayuda a unir usuarios con gustos similares. Esto se lo realiza mediante la valoración del ítem que se haya dado en la recomendación [17]. Este tipo de sistema de recomendación no necesita demasiada información del ítem, ya que el usuario es quien realmente brinda la información que se

toma en cuenta para la recomendación.

#### 1.3.4.2. Sistemas de recomendación basados en contenido

El sistema de recomendación basado en contenido no usa la valoración que el usuario brinda, si no, usa otros parámetros como: la información del producto, o el perfil del usuario. Así, usualmente mediante el análisis de estos dos parámetros se realiza la recomendación [17]. Este tipo de sistema se usa más en productos nuevos ya que usualmente se posee una gran información del producto y sus características.

#### 1.3.4.3. Sistemas de recomendación basados en conocimiento

Este tipo de sistema ayuda a resolver cuando un producto nuevo aparece y poder brindar una recomendación. Para poder entregar una recomendación, toman todo el conocimiento del producto y del usuario para así brindar una recomendación personalizada [19]. El usuario podrá controlar las recomendaciones que se le brindan mediante diferentes filtros. Este tipo de sistema ayuda a corregir las desventajas que tienen los sistemas anteriormente mencionados.

#### 1.3.4.4. Sistemas de recomendación híbridos

Los sistemas de recomendación híbridos usan una combinación de los diferentes tipos de sistemas recomendadores, con el fin de proporcionar un mejor rendimiento y presentar una mejor recomendación [19].

#### 1.3.4.5. Sistemas de recomendación demográfica

En este tipo de sistema, las características del usuario, como: género, edad, educación, etc., son consideradas para emitir una recomendación. No se necesitan calificaciones o reseñas proporcionadas por el usuario a los productos, lo que permite brindar una recomendación, aunque el usuario nunca haya dado alguna calificación a algún producto [18].

#### 1.3.4.6. Sistemas de recomendación basado en palabras clave

Los sistemas de este tipo se basan en medir las preferencias del usuario basado en palabras clave. Se utiliza para ello el análisis de textos escritos por los usuarios para generar recomendaciones. Para este tipo de sistema de recomendación se clasifica para el usuario como anterior y activo. Del usuario anterior se extraen de sus reseñas un conjunto de palabras clave que son almacenadas dentro de una base de datos. Así cuando el usuario activo proporciona una nueva palabra clave y la importancia ponderada de ésta, se calcula la similitud de las palabras clave de las revisiones del usuario anterior con la palabra clave dada por el usuario activo. Los métodos más

comunes utilizados para calcular la similitud entre las palabras clave son el modelo de Jaccard, NGram y Vector Space [18].

### 1.3.5. Ventajas y desventajas de los sistemas de recomendación

**Tabla 1.2:** Ventajas y desventajas de los sistemas de recomendación [17] [18] [20]. Elaborado por autores.

| Tipo | Ventajas | Desventajas |
|---|---|---|
| **Filtrado colaborativo** | • No se necesita información acerca de los productos<br>• Las clasificaciones se pueden utilizar para proporcionar las recomendaciones | • Problema de arranque en frío<br>• Alto costo para encontrar el mejor vecino<br>• Problema de las ovejas negras<br>• Problema de escasez de datos<br>• Escalabilidad<br>• Calidad |
| **Basado en contenido** | • Brinda recomendaciones tan pronto se tenga información de los productos disponibles | • Problema de casualidad |
| **Basado en conocimiento** | • Solventa problemas con sistemas de filtrado colaborativo, como: calidad y costo de encontrar el mejor vecino<br>• Mejora problemas de casualidad<br>• Facilita el arranque en frío | • Reglas de asociación entre los productos y bases de conocimiento<br>• Complejidad crece a medida del número de productos |
| **Híbridos** | • Aumento de la precisión<br>• Mejora el rendimiento<br>• Superar los problemas de sistema de recomendación basado en filtrado colaborativo y el sistema de recomendación basado en el contenido.<br>• Supera problemas de los sistemas de filtrado colaborativo como: ovejas negras y escasez de datos | • Sistemas complejos<br>• Caros de implementar |
| **Demográfica** | • Mejor recomendación de calidad para el usuario | • Problema de arranque en frío |
| **Basado en palabras clave** | • Puede manejar comentarios en forma de revisiones de texto<br>• Puede integrarse con las redes sociales<br>• Puede incorporar calificación multicriterio<br>• Buena precisión | • Difícil hacer cálculo de similitud<br>• Clasificación de palabras clave<br>• Calculo el peso |

## 1.4. HERRAMIENTAS DE DESARROLLO

Para la implementación del sistema de recomendación se usaron las siguientes herramientas:

**Tabla 1.3:** Herramientas utilizadas para el desarrollo. Elaborado por autores.

| Nombre | Descripción | Aplicado en |
|---|---|---|
| python | Python es un lenguaje de programación que permite trabajar más rápidamente e integrar sus sistemas de manera más efectiva [21]. | Sistema de recomendación |
| ANACONDA | Anaconda es un distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático [22]. | Sistema de recomendación |
| CONDA | Conda es un sistema de gestión de paquetes de código abierto y de gestión del entorno [23]. | Sistema de recomendación |
| jupyter | Jupyter Notebook es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo [24]. | Sistema de recomendación |
| Scrapy | Scrapy es un marco rápido de rastreo y raspado web de alto nivel, utilizado para rastrear sitios web y extraer datos estructurados de sus páginas [25]. | Recopilación de información |
| Lucidchart | Lucidchart es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas y muchos otros tipos de diagrama [26]. | Creación de ilustraciones |
| GitHub | GitHub es un repositorio para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador [27]. | Sistema de recomendación |
| GitKraken | GitKraken es una herramienta multiplataforma que presenta una interfaz gráfica para la gestión de repositorios git [28]. | Sistema de recomendación |
| Visual Studio Code | Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio [29]. | Sistema de recomendación |

# 2. METODOLOGÍA

Para el proyecto en mención se aplicará la metodología Scrum. Para la realización del proyecto y para el cumplimiento de los objetivos descritos, se realizaron las siguientes actividades principales:

- Recolección de información.
- Análisis y clasificación de información.
- Implementar un sistema de recomendación con los ratings.
- Pruebas con las recomendaciones brindadas por el sistema de recomendación.

## 2.1. DESCRIPCIÓN DE SCRUM

SCRUM establece un número de recomendaciones ayudan administrar y controlar los procesos de desarrollo de proyectos que implican un gran volumen de cambios rápidos. SCRUM en un marco ágil y liviano que resulta básicamente de la unión del modelo iterativo y el modelo incremental, debido a que todas las iteraciones del proyecto son sucesivas e interactivas [30]. En la Figura 2.1 se observa el proceso que realiza SCRUM [31].

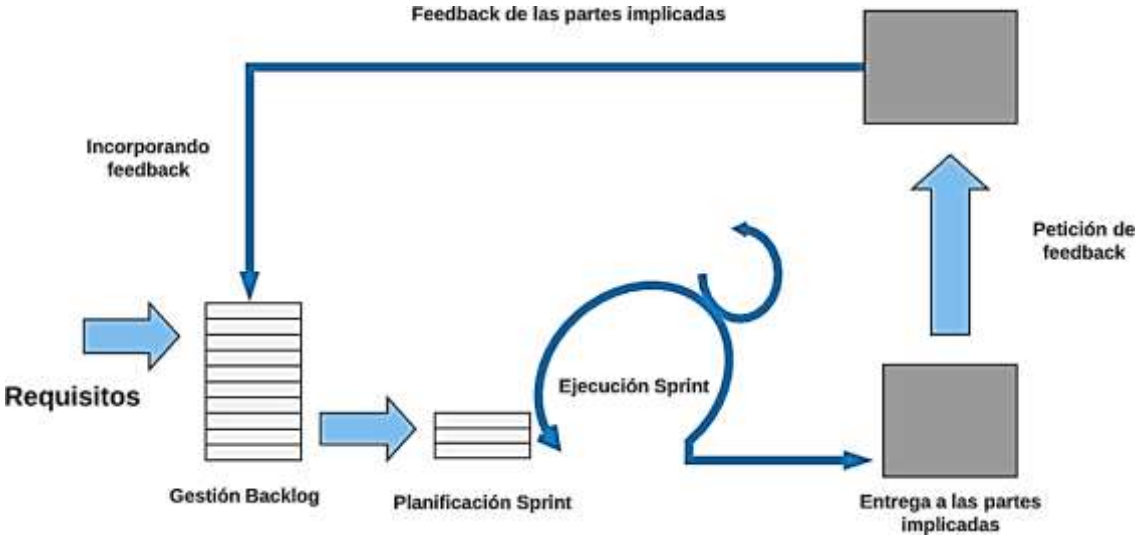



**Figura 2.1:** Proceso de SCRUM [30]. Elaborado por autores.

SCRUM cuenta con un ciclo de vida, el cual está compuesto por las siguientes etapas (Figura 2.2) [32]:

- **Planning:** Visión y expectativas correctas del proyecto, además, la financiación que se requiere al momento de desarrollar el proyecto.
- **Staging:** Identificar los requisitos del proyecto y priorizar las iteraciones para el desarrollo de este.

- **Development:** Implementación del proyecto.
- **Release:** Entrega y despliegue del proyecto desarrollado.

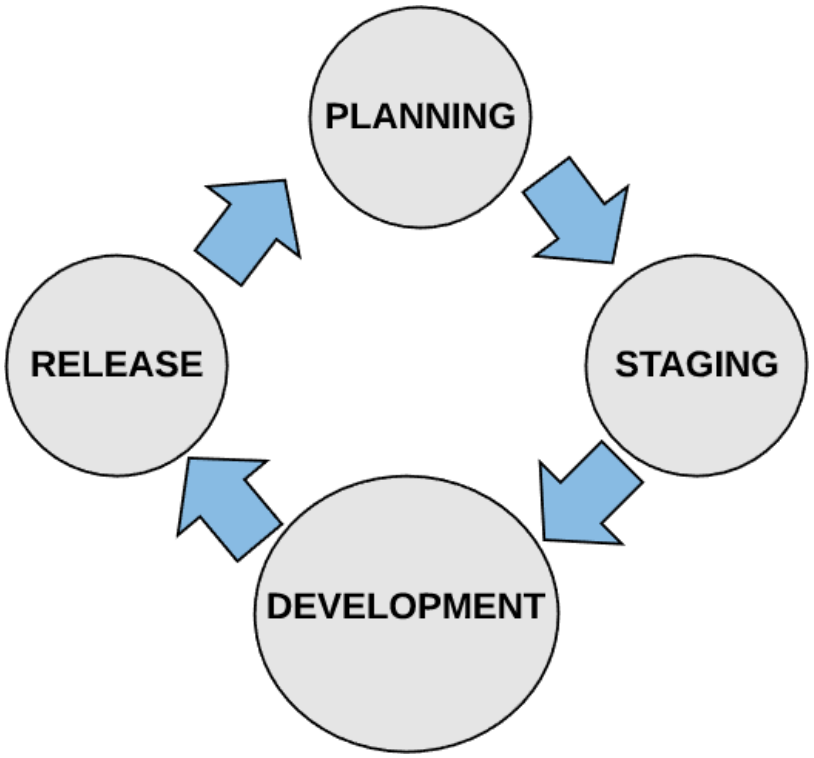



**Figura 2.2:** Ciclo de vida de SCRUM [33]. Elaborado por autores.

SCRUM agrupa las actividades en tres fases principales las cuales tiene el nombre de *Pregame, Game y Postgame*.

- ***Pregame,*** es la fase inicial, donde el proyecto se encuentra definido apenas por la visión de este y donde el proyecto no se encuentra muy bien definido. Conforme se avance se irá afinando el proyecto mediante los respectivos Sprint [32]. La visión del proyecto es responsabilidad del propietario del proyecto quién deberá preparar una lista de prioridad de requisitos funcionales y no funcionales del proyecto. Esta lista se denomina **Product Backlog**, e incluye la estimación en tiempo y costo, así como la fecha de entrega final y sus lanzamientos previos a la entrega final [32].
- ***Game***, es la fase en la cual se comienza a desarrollar el proyecto mediante iteraciones pequeñas las cuales son dominadas Sprint [32]. Cada Sprint cuenta con un tiempo determinado, su intervalo de tiempo depende de la complejidad y el riesgo del proyecto que se encuentra en desarrollo. Cada Sprint puede durar de uno a cuatro semanas.
- ***Postgame***, es la fase de cierre del proyecto, en la cual las se han implementado todas las características descritas en el Product Backlog durante la fase de

desarrollo del proyecto. El proyecto está cerrado cuando se cumplieron todos los objetivos en los que se plantearon en la fase de *Pregame* [32].

En cada Sprint el grupo de trabajo se encuentran realizando las tareas encomendadas en un tiempo determinado bajo la supervisión del ***Scrum Master***. Las tareas se van acumulando de acuerdo con el Product Backlog. El Product Backlog contiene una lista de requisitos del propietario del proyecto, las cuales se convierten en historias de usuarios. Sin embargo, si algún Sprint se atrasó por algún motivo, para el siguiente Sprint se incluirán las tareas no culminadas. Al final del día se deberá realizar un ***Daily Scrum***, el cual consiste en informar los avances que sean realizados en las tareas del Sprint. Cuando se termine un Sprint se le debe entregar un producto funcional al propietario del proyecto. El propietario verificará cada tarea que se realizó en el Sprint, y así poder ver el incremento del proyecto [30].

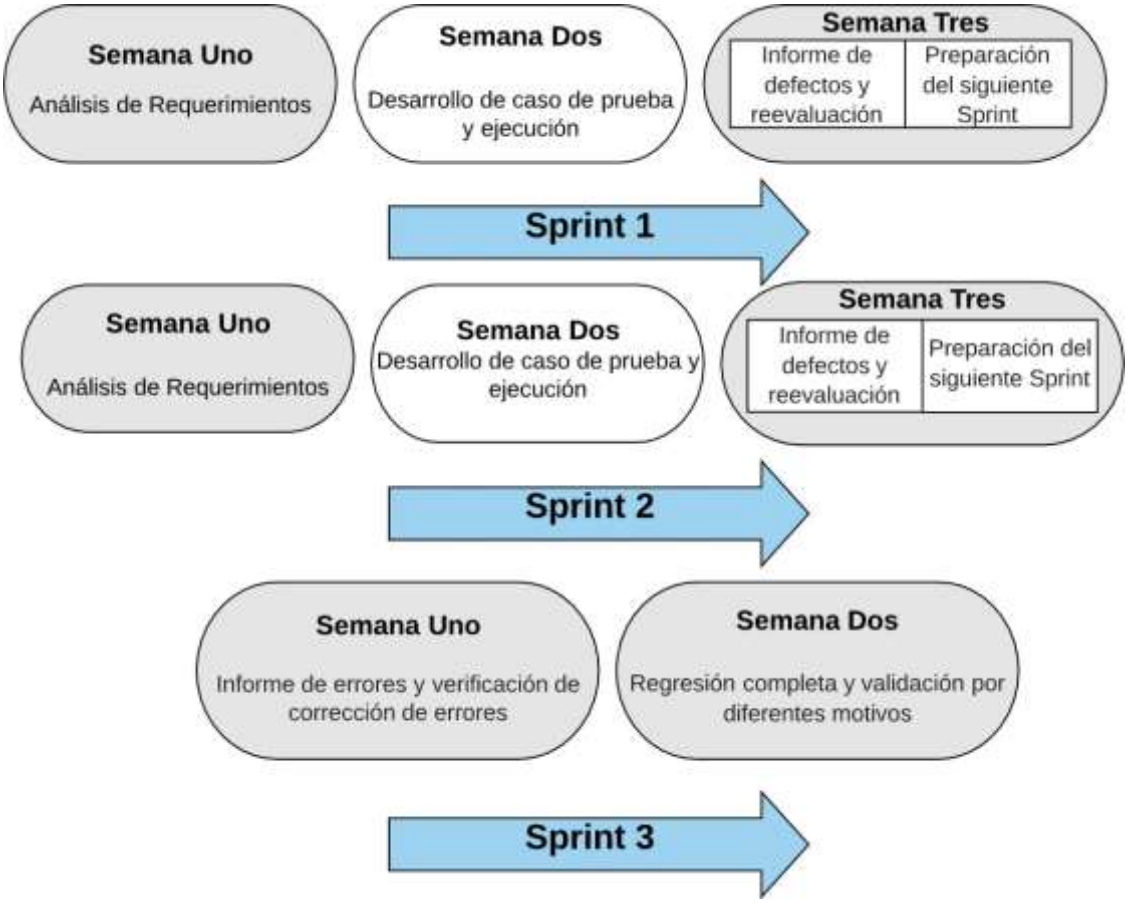



**Figura 2.3:** Ciclo de Sprint en Scrum [30]. Elaborado por autores.

Las etapas de Sprint se encuentran descritas en la Tabla 2.1:

**Tabla 2.1:** Etapas Sprint. Elaborado por autores.

| Etapa | Descripción |
|---|---|
| **Sprint Planning** | Se planifica las actividades que se van a realizar durante el Sprint, y cómo se van a realizar. El equipo otorga prioridades a las tareas y evalúa el porcentaje que va a poder realizar y la duración del Sprint. |
| **Daily Scrum** | Es una reunión diaria la cual dura máximo 15 minutos, en donde, el equipo comenta los avances o inconvenientes que han tenido durante la realización de las tareas. |
| **Sprint Development** | Es la etapa, donde, se realiza el diseño, desarrollo y prueba; para cada tarea dependiendo de la prioridad que se le otorgo. |
| **Sprint Review** | Es la etapa donde, el equipo y el Product Owner realizan una revisión a las tareas del Sprint que se realizaron, las que no se culminaron y las que no se realizaron. Además, se realiza comentarios sobre la funcionalidad de las tareas realizadas, los cuales se tomarán en cuenta en el siguiente Sprint. |
| **Sprint Retrospective** | El equipo se reúne, discute sobre los comentarios que se dieron en la etapa de Sprint Review, la comunicación, el proceso y que se hizo bien o mal durante el Sprint. |

SCRUM define tres tipos de roles que son los siguientes:

- **Product Owner**

  Es el dueño del producto. Realiza la lista de prioridades, donde, él es el encargado de ver que el producto cumpla su propósito con se debe, también, es responsable de las ganancias o pérdidas del producto. Trabajado con el Equipo SCRUM, sin embargo, únicamente solo puede haber un Product Owner.

- **Scrum Master**

  Es el líder del equipo Scrum, donde, pone en práctica SCRUM, sus reglas y valores. Realiza el Daily Scrum, donde, se informará del progreso del producto. También, es el responsable de que exista un buen ambiente entre el equipo. Debe estar presente en todas las etapas del Sprint de SCRUM.

- **Team**

  El equipo está conformado con hasta 8 miembros. Dentro del equipo no existen roles específicos. Los miembros del equipo deciden como se va a dividir las tareas a desarrollar. Cada miembro debe tener habilidades de diseño, desarrollar, pruebas y documentación.

## 2.2. PROYECTO SCRUM

### 2.2.1. Investigación Preliminar

Para resolver el problema que se planteó en el presente trabajo, se analizó el uso de los sistemas de recomendación en sus diferentes aplicaciones. De igual manera, se hizo

una revisión de las tareas y problemas que tiene un analista de ciberseguridad.

### 2.2.2. Roles

L os roles son impuestos por la metodología SCRUM se pueden visualizar en la Tabla 2.2

**Tabla 2.2:** Roles SCRUM.

| Rol | Encargado |
|---|---|
| Product Owner | PhD. Edison Loza Aguirre |
| Scrum Master | Carlos Ayala Tipán |
| Research Team | Carlos Ayala Tipán y Kevin Jimenez Saraguro |

### 2.2.3. Requerimientos

El Sistema de recomendación para ciberseguridad a desarrollarse es un sistema de recomendaciones en el cual está basado en las anomalías y sus respectivas contramedidas. Su finalidad es ayudar a los analistas de ciberseguridad de cualquier empresa o medio a reducir el tiempo de mitigación de la anomalía. Los datos tendrán una estructura especifica en la cual se especificará el rating, el nombre del ataque, y su mitigación. Además, el sistema de recomendación estará basado en la calificación o rating de los usuarios para remendar la mitigación del ataque ingresado.

El sistema tendrá las siguientes capacidades:

- El ingreso del ataque por su nombre
- La visualización de las 5 mejores recordaciones. Los campos por visualizar son el nombre del ataque, descripción, su mitigación y su rating.

### 2.2.4. Product Backlog

El Product Backog se define en la Tabla 2.3, en él se listan las historias de usuario con la respectiva funcionalidad del sistema a cumplir.

**Tabla 2.3:** Lista de Historias de Usuarios. Elaborado por autores.

| Id Historia | Como un(a) | Yo quiero | Con el fin de | Horas |
|---|---|---|---|---|
| **UH-CS-1** | Investigador(a) | Recolectar la información de las anomalías, mitigaciones y sus respectivos ratings | Poder observar qué ataques se dan con más frecuencia y poder brindar una mejor recomendación. | 90 |
| **UH-CS-2** | Investigador(a) | Analizar la información recopilada | Que los datos recopilados sean verídicos y no contengan información innecesaria. | 80 |
| **UH-CS-3** | Investigador(a) | Clasificar la información previamente analizada y recopilada | Que la información esté clasificada por su nivel de criticidad, además que se pueda identificar claramente el nombre de la anomalía, una descripción corta y su posible mitigación | 40 |
| **UH-CS-4** | Investigador(a) | Investigar sobre los sistemas de recomendación. | Que el sistema de recomendación se acople a las necesidades de los analistas | 130 |
| **UH-CS-5** | Investigador(a) | Implementar un sistema de recomendación basado en rating. | Que mediante el rating que se haya otorgado se haga la recomendación. | 200 |
| **UH-CS-6** | Investigador(a) | Ingresar el nombre de la anomalía, por la cual se va a generar la recomendación. | Que el analista pueda ingresar libremente el ataque que él desea mitigar<br>sin ninguna restricción | 24 |
| **UH-CS-7** | Investigador(a) | Visualizar las 5 mejores recomendaciones para una anomalía. | Que el analista escoja la mejor recomendación basada en rating de otros usuarios. | 6 |
| **UH-CS-8** | Investigador(a) | Desarrollar una interfaz gráfica | Que el analista pueda visualizar de mejor manera tanto las anomalías como sus respectivas recomendaciones. | 40 |
| **UH-CS-9** | Investigador(a) | Encuestar a expertos. | Que el analista pueda brindar comentarios de uso del sistema de recomendación. | 90 |

### 2.2.5. Release planning

Una vez que se realizó el Product Backlog, a continuación, se planificó cuáles son las historias de usuario a realizar por cada Sprint. Para el desarrollo del producto, cada Sprint tendrá la duración de 2 a 5 semanas.

**Tabla 2.4:** Planificación Sprint. Elaborado por autores.

| Sprint 0 | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Sprint 6 | Sprint 7 |
|---|---|---|---|---|---|---|---|
| UH-CS-1 | UH-CS-2 | UH-CS-3 | UH-CS-4 | UH-CS-5 | UH-CS-6<br>UH-CS-7 | UH-CS-8 | UH-CS-9 |

### 2.2.6. Sprint 0

En la Tabla II.1 (Anexo II.1) se muestra la planificación de la ejecución de las fases del Sprint 0.

#### 2.2.6.1. Sprint Planning

##### *Sprint Goal*

Recopilar la información con la cual se va a construir el sistema de recomendación. La información recopilada será brindada por el CSIRT de la EPN, Symantec, OWASP, NIST y grupo de trabajo de la universidad de Trento-Italia. Esta última proporcionará el rating de cada anomalía con el cual se realizará la recomendación para el usuario.

##### *Sprint Backlog*

En la Tabla II.2 (Anexo II.1) se muestra la estimación del esfuerzo para el Sprint 0.

#### 2.2.6.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.4:** Ejecución del Sprint 0. Elaborado por autores.

#### 2.2.6.3. Inspección y adaptación

##### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 0. El objetivo para el Sprint 0 fue

cumplido con algunos inconvenientes, esto se debe a la criticidad y calidad de la información. Este inconveniente se debió a la inseguridad de las organizaciones a brindar información que les podría exponer. Sin embargo, se cumplieron los criterios de aceptación del producto. En la Tabla II.3 (Anexo II.1) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

#### *Actualización y refinamiento*

Se realizó la revisión del Sprint 0, donde se verificaron los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 0. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se vio afectado. De igual forma no se añadieron o modificaron historias de usuarios.

#### *Sprint Retrospective*

En el Burndown Chart del Sprint 0, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. Por la adquisición de los datos de anomalías de las diferentes entidades, se presentaron algunos contratiempos, retrasando algunas tareas planteadas. Una vez obtenida dicha información se continuo sin ningún inconveniente.

### 2.2.7. Sprint 1

En la Tabla II.4 (Anexo II.2) se muestra la planificación de la ejecución de las fases del Sprint 1.

#### 2.2.7.1. Sprint Planning

#### *Sprint Goal*

Analizar la información que fue recopilada. En este sprint se verificará que la información no contenga datos innecesarios y se modificará únicamente para que contenga lo necesario para que el sistema de recomendaciones funcione correctamente.

#### *Sprint Backlog*

En la

Tabla II.**5** (Anexo II.2) se muestra la estimación del esfuerzo para el Sprint 1.

#### 2.2.7.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.5:** Ejecución del Sprint 1. Elaborado por autores.

### 2.2.7.3. Inspección y adaptación

#### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 1. El objetivo para el Sprint 1 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.6 (Anexo II.2) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

#### *Actualización y refinamiento*

Se realizo la revisión del Sprint 1, donde se verificaron los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 1. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado. De igual forma. no se añadieron o modificaron historias de usuarios.

#### *Sprint Retrospective*

En el Burndown Chart del Sprint 1, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. No hubo contratiempos en el desarrollo de las tareas planteadas.

### 2.2.8. Sprint 2

En la Tabla II.7 (Anexo II.3) se muestra la planificación de la ejecución de las fases del Sprint 2.

### 2.2.8.1. Sprint Planning

#### *Sprint Goal*

Clasificar la información por medio de su nivel de criticidad y, además, que se identifique claramente el nombre de la anomalía, una descripción corta y su posible mitigación. También, se deberá contar con el rating de cada usuario, con ello se podrá brindar la recomendación basada en rating.

#### *Sprint Backlog*

En la Tabla II.8 (Anexo II.3) se muestra la estimación del esfuerzo para el Sprint 2.

### 2.2.8.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.6:** Ejecución del Sprint 2. Elaborado por autores.

### 2.2.8.3. Inspección y adaptación

#### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 2. El objetivo para el Sprint 2 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.9 (Anexo II.3) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

#### *Actualización y refinamiento*

Se realizo la revisión del Sprint 2, donde, se verificó los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 2. Los criterios fueron

cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado, de igual forma no se añadieron o modificaron historias de usuarios.

### *Sprint Retrospective*

En el Burndown Chart del Sprint 2, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. Al clasificar la información se encontró datos incoherentes, lo cual retraso las tareas presentadas, se solventó buscando en diferentes fuentes para completar y corregir la información.

## 2.2.9. Sprint 3

En la Tabla II.10 (Anexo II.4) se muestra la planificación de la ejecución de las fases del Sprint 3.

### 2.2.9.1. Sprint Planning

### *Sprint Goal*

Investigar sobre los sistemas de recomendación, su funcionamiento, cuáles son los diferentes sistemas recomendadores, sus características, ventajas y desventajas. Para con esto escoger la clase de sistema, el cual se acopla a nuestros requerimientos y poder implementarlo.

### *Sprint Backlog*

En la Tabla II.11 (Anexo II.4) se muestra la estimación del esfuerzo para el Sprint 3.

### 2.2.9.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.7:** Ejecución del Sprint 3. Elaborado por autores.

#### 2.2.9.3. Inspección y adaptación

##### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 3. El objetivo para el Sprint 3 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.12 (Anexo II.4) se muestra el resultado del Sprint Review de forma detallada, usando los criterios de aceptación establecidos para cada historia de usuario.

##### *Actualización y refinamiento*

Se realizo la revisión del Sprint 3, donde, se verifico los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 3. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado, de igual forma no se añadieron o modificaron historias de usuarios.

##### *Sprint Retrospective*

En el Burndown Chart del Sprint 3, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. No hubo contratiempos en el desarrollo de las tareas planteadas.

### 2.2.10. Sprint 4

En la Tabla II.13 (Anexo II.5) se muestra la planificación de la ejecución de las fases del Sprint 4.

#### 2.2.10.1. Sprint Planning

##### *Sprint Goal*

Construir un sistema de recomendación basado en rating, que leer los archivos que contienen la información para poder realizar la recomendación.

##### *Sprint Backlog*

En la
Tabla II.**14** (Anexo II.5) se muestra la estimación del esfuerzo para el Sprint 4.

#### 2.2.10.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.8:** Ejecución del Sprint 4. Elaborado por autores.

### 2.2.10.3. Inspección y adaptación

#### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 4. El objetivo para el Sprint 4 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.15 (Anexo II.5) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

#### *Actualización y refinamiento*

Se realizó la revisión del Sprint 4, donde, se verificó el cumplimiento los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 4. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado. De igual forma no se añadieron o modificaron historias de usuarios.

#### *Sprint Retrospective*

En el Burndown Chart del Sprint 4, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. No se presentó mayor inconveniente para su finalización.

### 2.2.11. Sprint 5

En la Tabla II.16 (Anexo II.6) se muestra la planificación de la ejecución de las fases del Sprint 5.

### 2.2.11.1. Sprint Planning

#### *Sprint Goal*

Al ingresar el ataque el cual se quiere mitigar, el sistema de recomendación nos brindará la recomendación con mejor calificación, y se podrá visualizar las mejores 5 recomendaciones.

#### *Sprint Backlog*

En la Tabla II.17 (Anexo II.6) se muestra la estimación del esfuerzo para el Sprint 5.

### 2.2.11.2. Ejecución del Sprint

A continuación, se presenta la ilustración de la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.9:** Ejecución del Sprint 5. Elaborado por autores.

### 2.2.11.3. Inspección y adaptación

#### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 5. El objetivo para el Sprint 5 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.18 (Anexo II.6) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

#### *Actualización y refinamiento*

Se realizo la revisión del Sprint 5, donde, se verificó los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 5. Los criterios fueron

cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado, de igual forma no se añadieron o modificaron historias de usuarios.

### *Sprint Retrospective*

En el Burndown Chart del Sprint 5, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. Brindar las mejores recomendaciones provocó un retraso medio en el sprint, sin embargo, no se presentó mayor complicación en cumplir todas las tareas.

## 2.2.12. Sprint 6

En la Tabla II.19 (Anexo II.7) se muestra la planificación de la ejecución de las fases del Sprint 6.

### 2.2.12.1. Sprint Planning

### *Sprint Goal*

Desarrollar una interfaz gráfica donde el usuario pueda interactuar con el sistema de recomendación, de esta forma pueda visualizar de mejor manera las anomalías y recomendaciones registradas.

### *Sprint Backlog*

En la Tabla II.20 (Anexo II.7) se muestra la estimación del esfuerzo para el Sprint 6.

### 2.2.12.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.10:** Ejecución del Sprint 6. Elaborado por autores.

#### 2.2.12.3. Inspección y adaptación

##### *Sprint Review*

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 6. El objetivo fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.21 (Anexo II.7) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

##### *Actualización y refinamiento*

Se realizó la revisión del Sprint 6, donde, se verificó los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 6. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado, de igual forma no se añadieron o modificaron historias de usuarios.

##### *Sprint Retrospective*

En el Burndown Chart del Sprint 6, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. Al principio se presentó un pequeño retraso, debido que el sistema estaba concebido para consola, pero al ver la necesidad del usuario final, se desarrolló una interfaz gráfica, en la cual la navegación sea más intuitiva.

### 2.2.13. Sprint 7

En la Tabla II.22 (Anexo II.8) se muestra la planificación de la ejecución de las fases del Sprint 7.

#### 2.2.13.1. Sprint Planning

##### *Sprint Goal*

Realizar encuestas a expertos acerca del funcionamiento y el nivel de satisfacción.

##### *Sprint Backlog*

En la Tabla II.23 (Anexo II.8) se muestra la estimación del esfuerzo para el Sprint 7.

#### 2.2.13.2. Ejecución del Sprint

A continuación, se presenta la ejecución del Sprint, la cual contiene las variaciones correspondientes a cada día del Sprint.


**Figura 2.11:** Ejecución del Sprint 7. Elaborado por autores.

#### 2.2.13.3. Inspección y adaptación

***Sprint Review***

Mediante la descripción de los criterios de aceptación, los cuales se definen para cada historia de usuario, se realizó la revisión el Sprint 7. El objetivo para el Sprint 7 fue cumplido sin ningún inconveniente, así como también, se cumplieron los criterios de aceptación del producto. En la Tabla II.24 (Anexo II.8) se muestra el resultado del Sprint Review de forma detallada usando los criterios de aceptación establecidos para cada historia de usuario.

***Actualización y refinamiento***

Se realizó la revisión del Sprint 7, donde, se verificó los criterios de aceptación de las historias de usuario que estaban planificadas para el Sprint 7. Los criterios fueron cumplidos sin ningún inconveniente, por tal motivo, el Product Backlog no se ve afectado, de igual forma no se añadieron o modificaron historias de usuarios.

***Sprint Retrospective***

En el Burndown Chart del Sprint 7, se muestra el esfuerzo realizado durante cada uno de los días de dicho Sprint. No se presentaron mayores inconvenientes que retrasen la finalización del Sprint.

## 2.3. EVALUACIÓN

### 2.3.1. Pruebas de caja negra

Para analizar el uso y comportamiento del sistema frente a los usuarios se previó la utilización de pruebas de caja negra. Nuestro objetivo fue diseñar pruebas, las cuales saquen a la luz los diferentes errores que puedan afectar la funcionalidad final. El siguiente

diagrama muestra el contexto de la prueba:

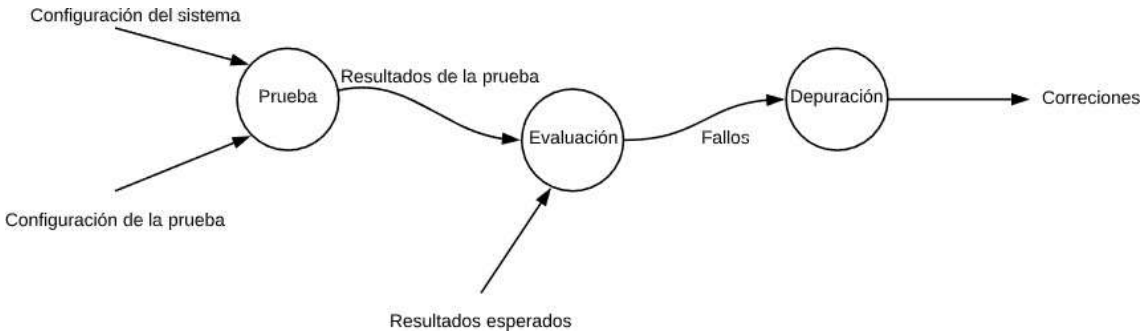



**Figura 2.12:** Contexto de funcionalidad de pruebas de caja negra. Elaborador por autores.

Las pruebas de caja negra, o también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Es decir, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa [34].


**Figura 2.13:** Prueba de caja negra. Elaborador por autores.

Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, es necesario tener en cuenta que en todo programa existe un conjunto de entradas que pueden causar un comportamiento erróneo, y como consecuencia producen una serie de salidas que revelan la presencia de defectos [35].

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes [34]:

1. Funciones incorrectas o faltantes,
2. Errores de interfaz,
3. Errores en las estructuras de datos o en el acceso a bases de datos externas,
4. Errores de comportamiento o rendimiento y,
5. Errores de inicialización y terminación.

# 3. RESULTADOS Y DISCUSIÓN

## 3.1. DESCRIPCIÓN DEL SISTEMA

### 3.1.1. Arquitectura del Sistema de recomendación

#### 3.1.1.1. Arquitectura Lógica

La base de conocimientos fue creada mediante Web Scraping, lo cual facilitó la recopilación de la información necesaria para la base de conocimiento. La fuente de la información recopilada se obtuvo de las paginas oficiales de OWASP, Symatec y SANS. Web Scraping ayudó a escanear el contenido de las páginas web para así poder obtener la información con la cual el sistema de recomendación va a ejecutarse

La información se dividirá en dos archivos:

1. En el primero se almacena la información recopilada de las páginas oficiales que consiste en el identificador del ataque, el nombre del ataque, la criticidad y la recomendación.
2. En el segundo se guarda el identificador del usuario que realizó la calificación de la recomendación y el identificador del ataque (el cual deberá ser idéntico al que tiene el archivo anterior)

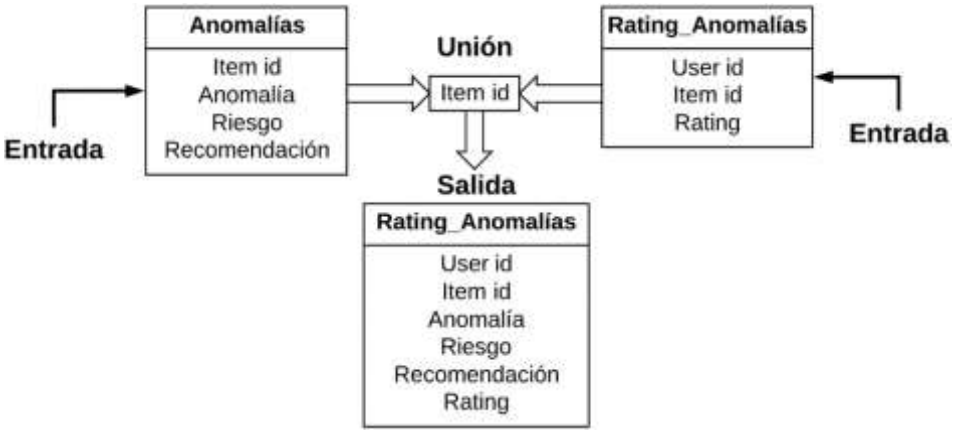



**Figura 3.1:** Estructura de la información. Elaborador por autores

Para la clasificación de la información recopilada se usa las librerías de Python que son las siguientes:

- NumPy
- Pandas

A través de estas, se puede limpiar y transformar la información para poder obtener información útil, la cual será utilizada en la implementación del sistema de recomendación. Esta arquitectura se puede apreciar en la Figura 3.2:

El funcionamiento del sistema es el siguiente:

1. La base de conocimientos (archivo uno) alimenta el motor de inferencia, el cual, contiene las reglas que servirán para clasificar la información recopilada.
2. Una vez rankeados las anomalías (archivo dos) se alimenta el sistema de recomendación, desarrollado en Python, este brinda las recomendaciones de respuesta a los incidentes que el usuario busque.
3. Para este tipo de sistema el usuario también es el experto en el dominio

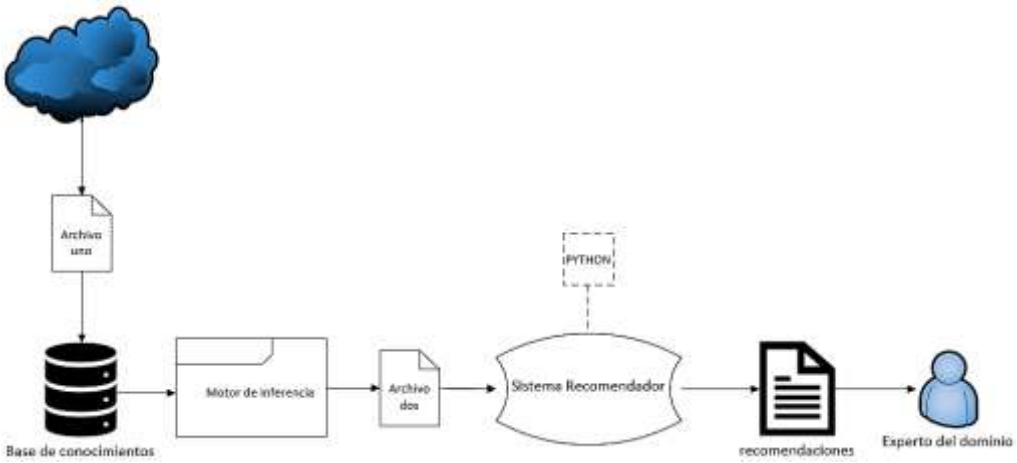



**Figura 3.2:** Arquitectura Lógica del sistema. Elaborador por autores.

### 3.1.1.2. Arquitectura Física

El sistema de recomendación es un software local que no necesita de ningún tipo de instalador y cuya información se almacena dentro del cliente que lo esté ejecutando. Para la recopilación de información, el sistema puede conectarse al internet para alimentar y almacenar la base de datos; una vez éstos estén almacenados de manera local el sistema, no se necesita ningún tipo de conexión a internet para su funcionamiento. Esta arquitectura se puede apreciar en la Figura 3.3:

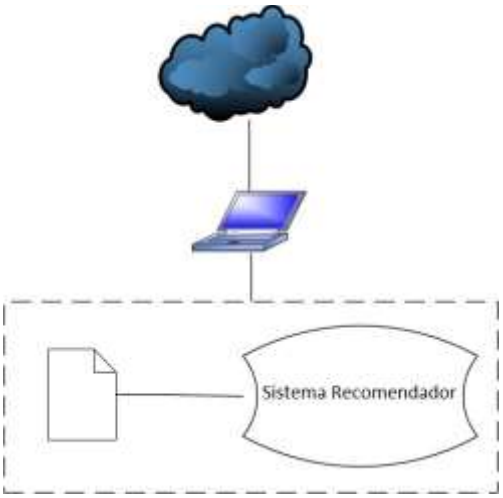



**Figura 3.3:** Arquitectura Física del sistema. Elaborador por autores.

El sistema de recomendación se ejecuta en una máquina virtual o física siempre que cumpla con las siguientes características mínimas.

- SO: Ubuntu 18.04 TLS
- CPU: 1
- Memoria: 2GB
- HDD: 10GB

Los requerimientos de la máquina son bajos, debido, a que el sistema de recomendación es bastante ligero y portable.

### 3.1.2. Descripción de los artefactos

#### 3.1.2.1. Sistema de Recomendación

*Pantalla inicio del sistema: En esta pantalla el analista podrá ingresar el nombre de la anomalía a la cual desea generar las recomendaciones*.

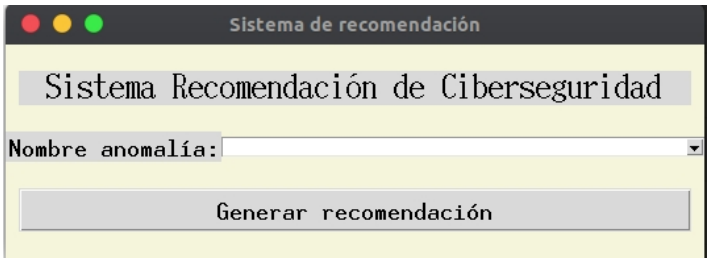



**Figura 3.4:** Ingreso de la anomalía. Elaborador por autores.

Para la búsqueda, el analista puede hacer uso de una lista desde donde puede visualizar las diferentes anomalías registradas.

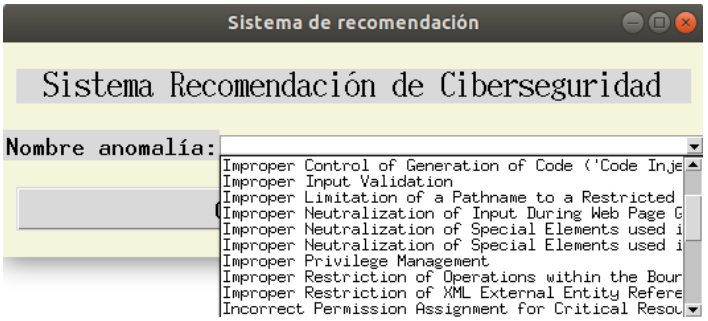



**Figura 3.5:** Ingreso de la anomalía. Elaborador por autores.

Además, para una mayor facilidad de uso en la búsqueda, el nombre de la anomalía podrá ser autocompletada siempre y cuándo se encuentre registrada.


**Figura 3.6:** Mensaje informativo anomalía no registrada. Elaborador por autores.

De no estar registrada la anomalía el sistema mostrará un mensaje informativo.

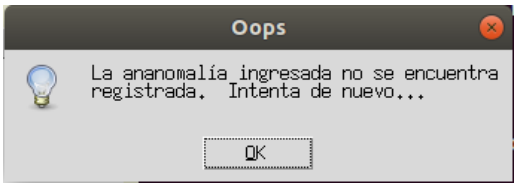



**Figura 3.7:** Autocomplete de la anomalía a buscar. Elaborador por autores.

*Pantalla anomalías similares*: En esta pantalla el analista podrá ver las cinco (5) anomalías similares (más cercanas) a la buscada, cuyas recomendaciones podrían ayudar a mitigar la anomalía buscada.


**Figura 3.8:** Los cinco ataques similares. Elaborador por autores.

*Pantalla Recomendación:* En esta pantalla el analista podrá ver la descripción y la o las recomendaciones con las cuales podrá mitigar la anomalía buscada.

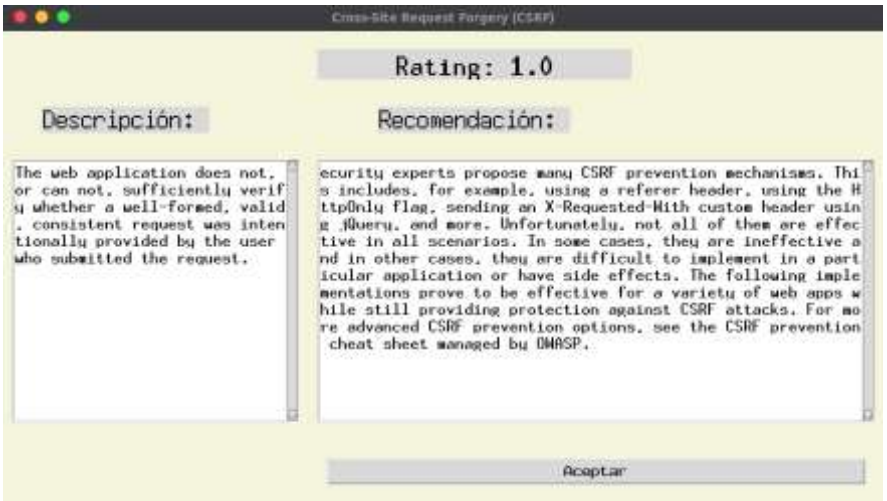



**Figura 3.9:** Descripción y recomendación para la anomalía buscada. Elaborador por autores.

## 3.2. RESULTADOS DE LA EVALUACIÓN

### 3.2.1. Resultados Prueba de caja negra

El diseño de las pruebas de caja negras se presenta a continuación. Estas se realizaron parar verificar el correcto funcionamiento del sistema sin tomar en cuenta el código fuente del sistema.

**Tabla 3.1:** Prueba de caja negra 01: Ingreso de la anomalía a ser mitigada.

| PCN 01 | Ingreso de la anomalía a ser mitigada |
|---|---|
| **Propósito** | Ingresar la anomalía a la cual el analista desea una recomendación, con la cual se podrá mitigar. |
| **Prerrequisito** | La anomalía debe estar registrada en la base de conocimientos. |
| **Datos de entrada** | Datos del formulario de la anomalía correctos. |
| **Pasos** | El analista debe ingresar el nombre de la anomalía a ser mitigada. |
| **Resultado Esperado** | Se visualiza las anomalías similares dependiendo del rating. |
| **Resultado Obtenido** | Se visualiza las anomalías similares con mejor rating. |
| **Resultado de prueba** | Correcto |

**Tabla 3.2:** Prueba de caja negra 02: Visualizar anomalías similares.

| PCN 02 | Visualizar anomalías similares |
|---|---|
| **Propósito** | Verificar que se encuentren anomalías similares a la que el analista desea mitigar, con su respectivo rating. |
| **Prerrequisito** | El ingreso del nombre de la anomalía a mitigar en el formulario. |
| **Datos de entrada** | Ninguno |
| **Pasos** | El analista pulsa en el botón de "Generar recomendación". |
| **Resultado Esperado** | El sistema muestra una lista con las anomalías similares con el mejor rating. |
| **Resultado Obtenido** | El sistema muestra una lista con las anomalías similares su rating y el número de rating. |
| **Resultado prueba** | Correcto |

**Tabla 3.3:** Prueba de caja negra 03: Visualizar la recomendación y su descripción.

| PCN 03 | Visualizar la recomendación y su descripción |
|---|---|
| **Propósito** | Visualizar la descripción y la recomendación de la anomalía seleccionada, con la cual se podrá mitigar la anomalía buscada. |
| **Prerrequisito** | El analista debe haber generado las recomendaciones. |
| **Datos de entrada** | Ninguno |
| **Pasos** | El analista pulsa dos (2) veces la anomalía. |
| **Resultado Esperado** | El sistema muestra una ventana con el rating, descripción y su respectiva recomendación para poder mitigar dicha anomalía. |
| **Resultado Obtenido** | La ventana se visualiza con el rating, descripción y su recomendación de mitigación. |
| **Resultado prueba** | Correcto |

### 3.2.2. Resultados Encuesta a expertos

Se realizaron encuestas a varios expertos tomando en cuenta los pilares fundamentales de TAM [9] para de esta manera medir la aceptación del sistema de recomendación.

La Figura 3.10 muestra el modelo de encuesta que se utilizó:

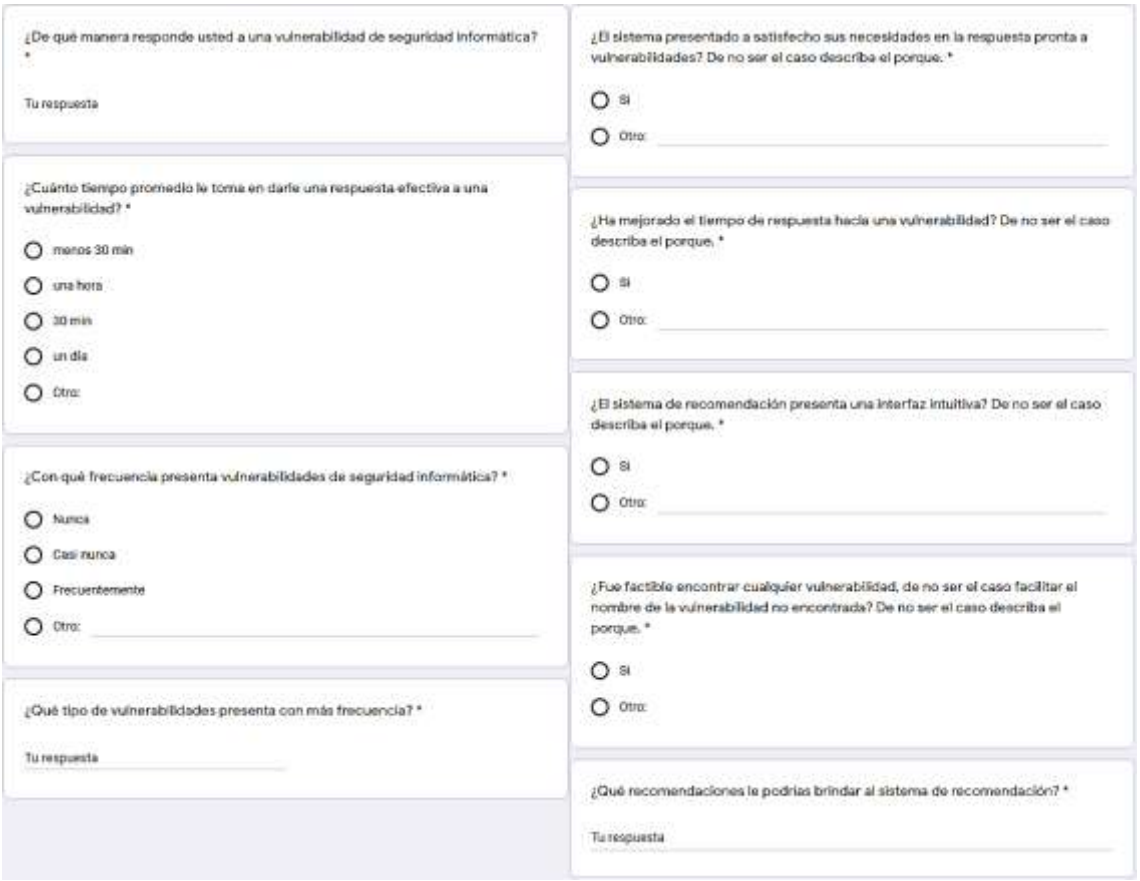



**Figura 3.10:** Modelo de encuesta. Elaborador por autores.

El sistema de recomendación fue evaluado por el siguiente grupo de expertos:

1. MSc. Roberto Andrade Docente Escuela Politécnica Nacional
2. PhD. Jenny Torres Docente Titular Escuela Politécnica Nacional
3. Ing. Ernesto Pérez Coordinador CSIRT-CEDIA
4. Ing. Javier Erazo Analista de Seguridad Informática CSIRT
5. MSc. Franklin Muñoz Gerente General PRISM Consulting Services

Los expertos fueron seleccionados con el objetivo de tener un amplio rango de criterios en la utilización y utilidad del sistema de recomendación [36]. De esta manera, se consideró a profesionales expertos en los campos de docencia, práctica y personas sin tanto conocimiento en el área de la seguridad informática; pertenecientes a sectores públicos y privados [36]. Los resultados de las encuestas fueron las siguientes:

### Utilidad Percibida

Los resultados encontrados han ayudado a evidenciar el comportamiento que tiene cada experto al brindar una repuesta ante las anomalías que se pueden presentar en sus

labores diarias y la manera en el que el sistema desarrollado les puede ser de utilidad. Así, cada uno de los expertos reaccionar de diversas formas ante una anomalía:

- Reportándola a la mesa de servicio.
- Tratar de resolver el momento que ocurre y buscar ayuda con partners dependiendo de la gravedad, aunque si se busca previamente que exista seguridades, pero siempre se limita al presupuesto de la empresa.
- Realizando el proceso de tratamiento de vulnerabilidades de acuerdo con un CSIRT.
- Realizando un análisis de vulnerabilidades, en el cual se detecta la presencia de dicha vulnerabilidad y se gestiona con el Administrador del Activo a fin de que implemente las medidas necesarias a fin de solventar la vulnerabilidad.
- Implementando un parche.

Dependiendo de la severidad de la anomalía, los expertos son capaces de brindar una respuesta en entre una hora y un día. En este lapso, no se contabiliza el tiempo extra que toma a cada experto realizar un informe para detallar la anomalía y la respuesta realizada.

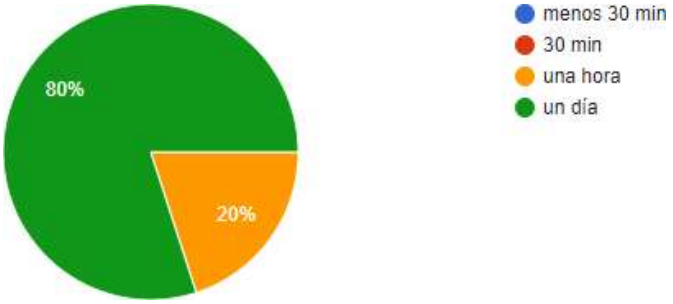



**Figura 3.11:** Resultados de tiempo promedio de respuesta efectiva. Elaborador por autores.

La mayoría de los expertos se ven confrontados frecuentemente a anomalías que pueden comprometer la seguridad informática.

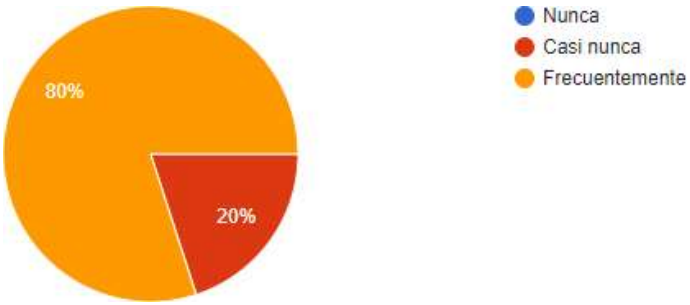



**Figura 3.12:** Resultados de frecuencia de presencia de anomalías. Elaborador por autores.

Dentro de las anomalías y vulnerabilidades a las cuales se enfrentan en su trabajo de los expertos, los expertos identificaron las siguientes:

- phishing
- virus, malware
- servicios desconfigurados
- vulnerabilidades de Inyección SQL
- vulnerabilidades de ventanas engañosas
- vulnerabilidades de Cross Site Scripting (XSS)
- aplicaciones obsoletas

Tres de cinco de expertos piensa que el sistema presentado a logrado satisfacer sus necesidades para brindar una respuesta pronta a las anomalías y vulnerabilidades. Los otros dos dieron una valoración positiva señalando mejoras que pueden introducirse.

El sistema de recomendación según tres de los expertos ha mejorado el tiempo de respuesta a anomalías y vulnerabilidades que puedan poner en riesgo la seguridad informática.

### Facilidad de uso

De cinco expertos, cuatro les fue factible encontrar cualquier anomalía o vulnerabilidad, la cual deseaban una recomendación.

Respecto a la interfaz del sistema de recomendación, tres expertos valoraron que es intuitiva y sencilla en el manejo y navegación. Los otros dos señalaron mejoras que pueden introducirse.

Algunos de los comentarios obtenidos con las encuestas son las siguientes:

- “El sistema debe estar en español para una mejor facilidad de uso para los analistas".
- “Se puede implementar una máquina inteligente la cual ayude con mayor efectividad a la clasificación de la información".
- “No esta intuitivo para el usuario final".
- “La información de la base de conocimiento debe estar en el idioma de origen de este".
- “El objetivo para poder dar uso a la herramienta es que se automatice.".

Las recomendaciones brindadas por los expertos son las siguientes:

- “Mejorar calidad de la interfaz respecto al aspecto gráfico".

- “Presentar al sistema de recomendación como una solución vía web".
- “Presentar una interfaz más intuitiva para el usuario final y personas que no son tan conocedoras en el tema de la seguridad informática".
- “El modelo escogido para el sistema de recomendación alivia las necesidades, sin embargo, podría aliviar más si se une con uno o varios modelos más para mejorar la efectividad".
- “El sistema resida en la nube para acceso desde cualquier lugar".
- “La base de conocimiento se actualice constantemente y que el sistema presente actualizaciones continuas".
- “Cada recomendación proporcionada por el sistema de recomendación debe tener su referencia bibliográfica o dar a conocer el sustento de que sea una recomendación fiable".

## 3.3. DISCUSIÓN

Según nuestros resultados, es necesario realizar una discusión del sistema recomendador seleccionado para esta investigación. En primer lugar, para un mejor enfoque en el tema de ratings se vio conveniente un tipo de sistema recomendador que involucre más al usuario. Por lo tanto, el modelo que mejor se ajustó a las necesidades, tanto del usuario como de los ítems (en este caso las anomalías y vulnerabilidades que comprometen la seguridad informática), fue un híbrido entre el sistema recomendador basado en filtro colaborativo y el sistema recomendado basado en el conocimiento. De esta manera, fue posible generar recomendaciones calificadas por un juicio de expertos y así aprovechar tanto el conocimiento como el dominio dentro del área de seguridad informática.

Además, este juicio de expertos sirvió para identificar las peores vulnerabilidades y anomalías dentro de las diferentes entidades a las cuales se les presentó el sistema. De esta manera, se pudo generar una base de conocimiento inicial que cuente con las peores vulnerabilidades encontradas y presentar las mejores recomendaciones para brindar una respuesta pronta y efectiva.

Sin embargo, dentro de cada entidad existe el recelo de presentar y compartir sus peores vulnerabilidades debido a la criticidad e impacto que puede repercutir que dicha información caiga en manos equivocadas. De esta manera, el juicio de expertos no siempre funcionó como se lo esperaba. Para solventar este inconveniente se utilizó un panel de expertos de diferentes fuentes.

También, se identificó que las recomendaciones dentro del grupo de sugerencias de anomalías y vulnerabilidades similares no siempre presentan afinidad con la que se busca solventar o entre ellas. Algunas de las recomendaciones a estas anomalías y vulnerabilidades no se ajustan a la realidad de las entidades a las cuales se les presentó el sistema.

Sin embargo, respecto a la hipótesis planteada dentro de la investigación, podemos señalar que se cumplió. De acuerdo con los resultados obtenidos, el sistema recomendador seleccionado ayuda a reducir el tiempo empleado, los procesos manuales y la subjetividad del analista de ciberseguridad. Esto se debe a que, al contar con una herramienta que consolida la información de la anomalía o de la vulnerabilidad a buscar con su o sus respectivas recomendaciones, facilita no solo la respuesta a los problemas en un tiempo menor sino, también a la realización de informes que sirven para documentar los mismos.

Se puede mejorar el tiempo de respuesta del sistema al unirlo otros tipos de sistemas recomendadores, donde además de involucrar al usuario se tome en cuenta a los ítems. De esta manera, se podrá mejorar las recomendaciones y el rating de los ítems dentro del sistema de recomendación. La combinación adecuada para un sistema híbrido, de acuerdo con los resultados, es el basado en conocimiento, basado en filtro colaborativo y el basado en palabras clave; para de esta manera incorporar calificación multicriterio.

# 4. CONCLUSIONES Y RECOMENDACIONES

## 4.1. CONCLUSIONES

- Las recomendaciones basadas en filtros colaborativo ayuda con la generación de la base de conocimiento con ayuda de un juicio de expertos. De esta manera fue posible obtener la información adecuada y necesaria que se almacenó en el sistema. Sin embargo, por sus principales desventajas con problemas de arranque en frío, la escasez y la calidad de datos, fue necesario utilizar recomendaciones basadas en conocimiento. Las recomendaciones basadas en el contenido se descartan debido a que desvincula a la valoración, la cual puede brindar el usuario restringiendo de esta manera los ratings para presentarle al usuario.
- Los expertos ocupados para la investigación son las entidades: Symantec, OWASP, NIST y un grupo de trabajo de la universidad de Trento-Italia. La información recopilada de sus datasets facilitó la elaboración de ratings, pues presentan datos bastante completos lo que permitió identificar las peores vulnerabilidades y para cada una de estas, las mejores recomendaciones.
- Los datasets recopilados permitieron la identificación de anomalías y vulnerabilidades comunes, las cuales se encuentran presentes dentro de las entidades. De esta manera, fue posible generar una base de conocimiento con información que al usuario final se le pueda presentar e interferir. Esta base de conocimiento toma en cuenta las calificaciones proporcionados por los usuarios y ayuda con la visualización de información consolidada de la anomalía o vulnerabilidad y de su o sus mejores recomendaciones.
- La implementación de un sistema de recomendación que facilite la visualización de información consolidada ayuda a mejorar el tiempo de respuesta, así como también, los procesos manuales y subjetividad del analista de ciberseguridad; pues al tener presente un juicio de expertos que alimenta la base de conocimientos, se les presenta información comprobada y veraz que facilita una mejor toma de decisión para dar una mejor respuesta.
- El sistema de recomendación de acuerdo con el sector donde se aplique presenta mejor utilidad, pues la información que se necesita presentar dentro de un sector público es más detallada que la información dentro de un sector privado. Esto se debe que el sector privado toma en cuenta más la resolución en un menor tiempo; a comparación del sector público, donde además de esto, se necesita documentar la descripción de la anomalía o vulnerabilidad solventada.

## 4.2. RECOMENDACIONES

- Una posible mejora del sistema de recomendación implementado dentro de esta investigación reside en la adaptación con un sistema basado en palabras clave para que de esta manera se pueda utilizar también recomendaciones multicriterio.
- La implementación del sistema de recomendación está pensada para ser offline y de manera local. Sin embargo, el sistema podría utilizarse únicamente donde se encuentre instalado, restringiendo el acceso por temas de ubicación geográfica. Para solventar esto, el sistema de recomendación se debe implementar dentro de una solución en la web para que el usuario final pueda dar respuestas pronto sin importar su ubicación geográfica.
- La actualización de la base de conocimiento también se debería realizar desde dentro de la interfaz para evitar problemas de inconsistencia de la información o problemas con la actualización de los archivos, los cuales contienen dichos datos.

# REFERENCIAS BIBLIOGRÁFICAS


[1] Western Governors University, «What does a cyber security analyst do?,» 13 Agosto 2018. [En línea]. Available: https://www.wgu.edu/blog/what-does-cybersecurity-analyst-do1808.html. [Último acceso: 2 Abril 2019].

[2] Bureau of Labor Statistics, «Information Security Analysts,» 12 Abri 2019. [En línea]. Available: https://www.bls.gov/ooh/computer-and-information-technology/mobile/information-security-analysts.htm. [Último acceso: 2 Abril 2019].

[3] A. Oltramari, N. Ben-Asher, L. Cranor, L. Bauer y N. Christin, «General Requirements of a Hybrid-Modeling Framework for Cyber Security,» *2014 IEEE Military Communications Conference,* 20 Noviembre 2014.

[4] I. Herwono y F. A. El-Moussa, «A System for Detecting Targeted Cyber-Attacks Using Attack Patterns,» *Springer International Publishing AG,* nº 867, pp. 20-34, 9 Junio 2018.

[5] C.-H. Hur, S.-P. Kim, Y.-S. Kim y J.-H. Eom, «Changes of Cyber-Attacks Techniques and Patterns after the Fourth Industrial Revolution,» *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW),* 21-23 Agosto 2017.

[6] J. B. Schafer, J. A. Konstan y J. Riedl, «E-Commerce Recommendation Applications,» *Data Mining and Knowledge Discovery,* vol. 5, pp. 115-153, Enero 2001.

[7] R. Burke, «Hybrid Recommender Systems: Survey and Experiments,» *User Modeling and User-Adapted Interaction,* vol. 12, pp. 331-370, Noviembre 2002.

[8] J. C. Gallardo, «Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía,» Jaén, 2012.

[9] F. D. Davis, «Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,» *MIS Quarterly,* vol. XIII, nº 3, pp. 319-340, Septiembre 1989.

[10] Public Safety Canada, «Canada's Cyber Security Strategy For a stronger and more prosperous Canada,» 2010.

[11] M. Lezzi, M. Lazoi y A. Corallo, «Cybersecurity for Industry 4.0 in the current

literature: A reference framework,» *Computers in Industry,* vol. 103, pp. 97-110, December 2018.

[12] CISCO, 2020. [En línea]. Available: https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html.

[13] E. Amoroso, Cyber Security, Primera ed., S. Press, Ed., New Jersey, 2006.

[14] Universidad Veracruz, 2020. [En línea]. Available: https://www.uv.mx/csirt/que-es-un-incidente-de-ciberseguridad. [Último acceso: Enero 2020].

[15] F. Isinkaye, Y. Folajimi y B. Ojokoh, «Recommendation systems: Principles, methods and evaluation,» *Egyptian Informatics Journal,* vol. 16, Agosto 2015.

[16] T. Mahmood y F. Ricci, «Improving recommender systems with adaptive conversational strategies,» *Proceedings of the 20th ACM conference on Hypertext and hypermedia,* June 2009.

[17] M. C. Martínez, «Sistemas de Recomendación basados en técnicas de predicción de enlaces para jueces en línea,» Madrid, 2017.

[18] N. Vaidya y A. R. Khachane, «Recommender systems-the need of the ecommerce ERA,» *2017 International Conference on Computing Methodologies and Communication (ICCMC),* Julio 2017.

[19] H. Casanova, E. Ramos y H. Nuñez, «Sistema Basado en Conocimiento para Recomendación de Información Turística Venezolana,» *III Simposio Científico y Tecnológico en Computación - SCTC 2014,* Mayo 2014.

[20] P. Pérez, «Recomendaciones en tiempo real mediante filtrado colaborativo incremental y real-time Big Data,» Madrid, 2016.

[21] Python, 2020. [En línea]. Available: https://www.python.org/about/. [Último acceso: Enero 2020].

[22] Anaconda, 2020. [En línea]. Available: https://www.anaconda.com/why-anaconda/. [Último acceso: Enero 2020].

[23] Conda, 2020. [En línea]. Available: https://docs.conda.io/en/latest/. [Último acceso: Enero 2020].

[24] Jupyter, 2020. [En línea]. Available: https://jupyter.org/. [Último acceso: Enero 2020].

[25] Scrapy, 2020. [En línea]. Available: https://docs.scrapy.org/en/latest/. [Último acceso: Enero 2020].

[26] Lucidchart, 2020. [En línea]. Available: https://www.lucidchart.com/pages/es. [Último acceso: Enero 2020].

[27] GitHub, 2020. [En línea]. Available: https://guides.github.com/features/pages/. [Último acceso: Enero 2020].

[28] GitKraken, 2020. [En línea]. Available: https://support.gitkraken.com/. [Último acceso: Enero 2020].

[29] Visual Studio Code, 2020. [En línea]. Available: https://code.visualstudio.com/docs. [Último acceso: Enero 2020].

[30] A. Srivastava, S. Bhardwaj y S. Saraswat, «SCRUM model for agile methodology,» *International Conference on Computing, Communication and Automation (ICCCA2017),* Mayo 2017.

[31] K. M. Toapanta, «Método Ágil Scrum, aplicado a la implantación de un sistema informático para el proceso de recolección masiva de información con Tecnología Móvil,» Sangolquí, 2012.

[32] F. Anwer, S. Aftab, S. S. Muhammad y U. Waheed, «Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum,» *nternational Journal of Computer Science and Telecommunications,* vol. VIII, nº 2, pp. 1-7, Marzo 2017.

[33] E. Neelima y N. D. Saile, «A Study on SCRUM Agile Methodology And Its Knowledge Management Process,» *The International Journal Of Engineering And Science (Ijes),* vol. II, nº 3, pp. 22-27, 2017.

[34] R. S. Pressman, Ingeniería de Software Un enfoque práctico, Séptima ed., McGraw-Hill, 2010, p. 423.

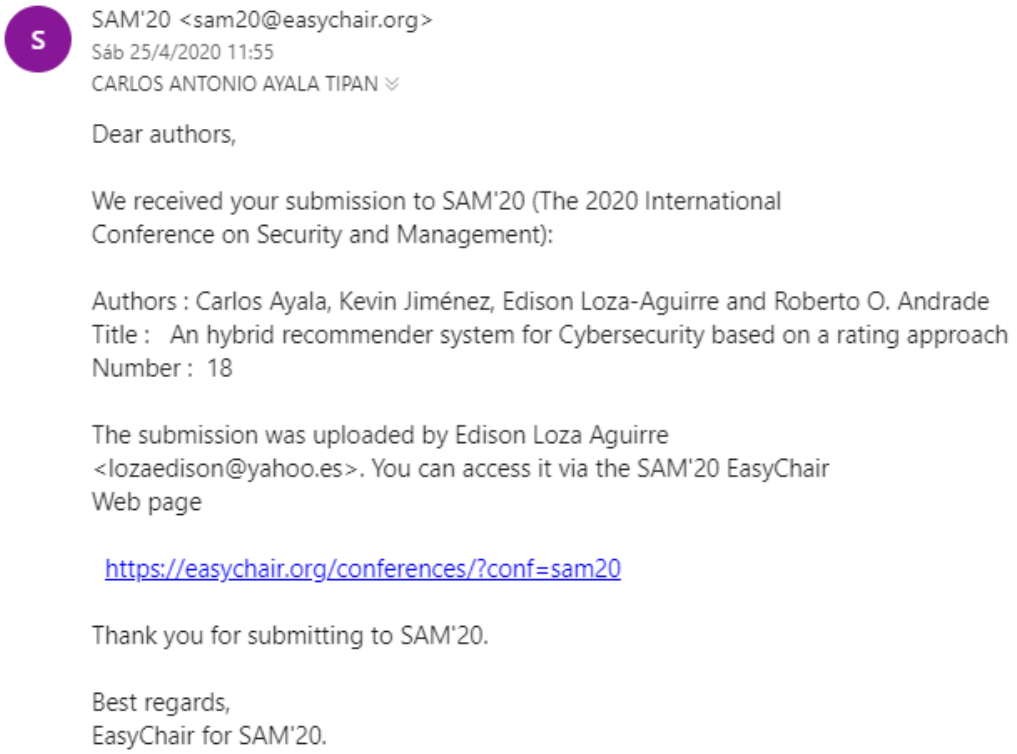
[35] Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla, «Técnicas de Evaluación Dinámica».

[36] E. Loza Aguirre y A. Buitrago Hurtado, «Qualitative Assessment of User Acceptance within Action Design Research and Action Research: Two Case Studies,» *Latin-American Journal of Computing - LAJC,* vol. I, nº 1, pp. 7-16, 17 Octubre 2014.

# ANEXO

# ANEXO I. ARTÍCULO CIENTÍFICO

## I.1 NOTIFICACIÓN DE RECEPCIÓN POR PARTE DE SAM'20



SAM'20 <sam20@easychair.org>
Sáb 25/4/2020 11:55
CARLOS ANTONIO AYALA TIPAN

Dear authors,

We received your submission to SAM'20 (The 2020 International Conference on Security and Management):

Authors : Carlos Ayala, Kevin Jiménez, Edison Loza-Aguirre and Roberto O. Andrade
Title : An hybrid recommender system for Cybersecurity based on a rating approach
Number : 18

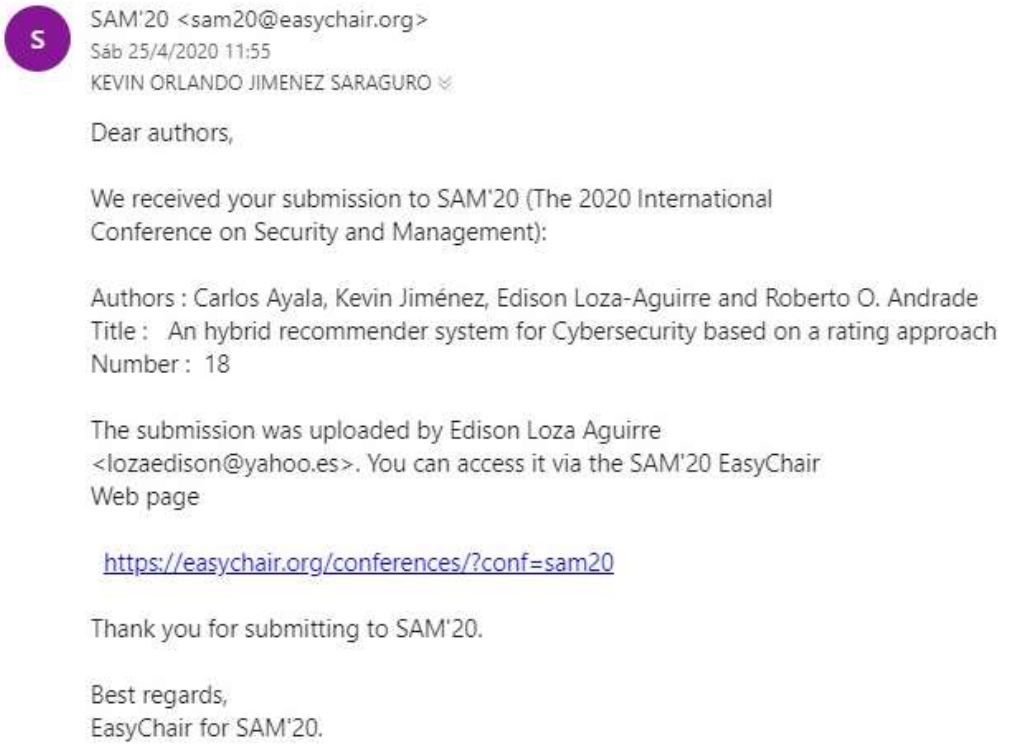
The submission was uploaded by Edison Loza Aguirre <lozaedison@yahoo.es>. You can access it via the SAM'20 EasyChair Web page

https://easychair.org/conferences/?conf=sam20

Thank you for submitting to SAM'20.

Best regards,
EasyChair for SAM'20.

**Figura. I.1.** Notificación de recepción a carlos.ayala01@epn.edu.ec



SAM'20 <sam20@easychair.org>
Sáb 25/4/2020 11:55
KEVIN ORLANDO JIMENEZ SARAGURO

Dear authors,

We received your submission to SAM'20 (The 2020 International Conference on Security and Management):

Authors : Carlos Ayala, Kevin Jiménez, Edison Loza-Aguirre and Roberto O. Andrade
Title : An hybrid recommender system for Cybersecurity based on a rating approach
Number : 18

The submission was uploaded by Edison Loza Aguirre <lozaedison@yahoo.es>. You can access it via the SAM'20 EasyChair Web page

https://easychair.org/conferences/?conf=sam20

Thank you for submitting to SAM'20.

Best regards,
EasyChair for SAM'20.

**Figura. I.2.** Notificación de recepción a kevin.jimenez@epn.edu.e

## I.2 ARTÍCULO ENVIADO A SAM'20

# An hybrid recommender system for Cybersecurity based on a rating approach


Carlos Ayala
*Facultad de Ingeniería en Sistemas*
*Escuela Politécnica Nacional*
Quito, Ecuador
carlos.ayala01@epn.edu.ec

Kevin Jiménez
*Facultad de Ingeniería en Sistemas*
*Escuela Politécnica Nacional*
Quito, Ecuador
kevin.jimenez@epn.edu.ec

Edison Loza-Aguirre
*Departamento de Informática y Ciencias de la Computación*
*Escuela Politécnica Nacional*
Quito, Ecuador
edison.loza@epn.edu.ec

Roberto O. Andrade
*Departamento de Informática y Ciencias de la Computación*
*Escuela Politécnica Nacional*
Quito, Ecuador
roberto.andrade@epn.edu.ec



**Track:**

The 2020 International Conference on Security and Management (SAM'20)
Security Management



**Contact Author:**

Edison Loza-Aguirre, edison.loza@epn.edu.ec



**Keywords:**

Security operations, recommendation system, collaborative filtering, security analysis, knowledge base.


**Figura. I.3.** Página carátula de artículo científico

# A hybrid recommender system for Cybersecurity based on a rating approach

***Abstract*— The main function of a security analyst is to protect and make the best decisions for preserving the integrity of computer systems within an organization. Every day, they must deal with many anomalies and attacks. To provides a quick response, the analyst usually depends on his good judgement, which should lead him to execute manual processes in a limited time. By dealing with too many anomalies and vulnerabilities, responses are only provided to those threats with the highest level of criticality. This research aims to propose a tool that would help analysts to filter out anomalies, vulnerabilities, and latent risks. To meet this objective, a recommendation system based on collaborative filtering and knowledge was developed, generating ratings of the worst cases with the best recommendations based on expert judgement. During tests, the system allowed an improvement in the response time from analysts to solve problems. It also eliminated the subjectivity of the analyst and reduced the number of manual processes.**

***Keywords*— *security operations, recommendation system, collaborative filtering, security analysis, knowledge base.***

## I. Introduction

The cybersecurity analysts, according to Randall Fietzsche [1], are the professionals in charge of analysing the risk and threats that may compromise an organization, and then plan and execute security measures with the aim of protecting the organizational networks and computer systems [2]. In other words, their job is to help the organization understand what is happening and where it should go in terms of computer security [2].

Cybersecurity analysts' work involves dealing with risks, vulnerabilities, and threats on a daily basis, leading them to search for a frame of reference that allow them to prioritize the most critical incidents and attacks in order to get the best actions to counter them.

However, there are three factors that can affect their decisions [2]: (1) Time, because cybersecurity analysts must resolve attacks as soon as possible (2), 'Manual processes and methodologies', because most of the process to identify and respond to attacks are manual and (3) the 'Analyst subjectivity', because the analyst usually depends on his good judgement and experience at the time to make decisions and perform the tasks to solve a security incident. These three factors can affect the performance of any cybersecurity analyst regardless of the environment in which they work.

Above all, if we consider that traditional practices tend to report large numbers of alerts, which should be examined and verified with all the information available, be it structured or not [3]. This makes an analyst feel overwhelmed when trying to discriminate which product, content or service meets the correct need to optimally solve a problem [4]. It is in this context that recommendation systems can be used to help them in their search for solutions. A recommendation system is the one that produces personalized recommendations as output, or it can guide users to choose interesting or useful products in line with their needs [5]. In our case, we refer as a 'product' to all information piece about a cybersecurity incident. The development of such a system in cybersecurity context would alleviate the tasks and problems that a cybersecurity analyst can present.

In this research, we propose a recommendation system that seeks to prioritize vulnerabilities, threats and risks; and the possible solutions to them. The aim is to improve the analyst's response time to different incidents. The system will

**Figura. I.4.** Página uno de artículo científico

provide the best responses for each classified rating-based security incident with the help of experts in the area.

This article is organized as follows. Section II presents the theoretical background about security incident response processes and recommendation systems. Section III shows the process followed for implementing the recommendation system. Section IV presents the developed system. In Section V we offer the results obtained from the evaluation with experts. Section VI shows a brief discussion of the results obtained. Finally, in Section VII we highlight our conclusions.

## II. Theoretical Background

### A. Cybersecurity Incident

The objective of digital attacks is to be able to access, modify or delete information. This is particularly challenging today because there are more devices than people, and attackers are becoming more innovative [6]. Cybersecurity refers to the way to protect information from any digital attack. It focuses on providing defensive methods to detect and capture any intruder who wants to compromise any information system [7].

A cybersecurity incident is an unwanted or unexpected event or set of events, which negatively impacts the processes and operations of organizations. Its impacts include disabling the use of information or the elimination or modification of data by corrupting information systems through malware infections, phishing, etc. [8].

### B. Recommendation Systems

Recommendation systems, also known as recommenders, are software techniques and tools that provide, as suggestions, a subset of elements belonging to a universe of alternatives that are considered the most appropriate for a user. In this sense, a recommendation system is considered as a decision-making support [9] [10].

One of the fundamental pillars of recommendation systems is the large amount of information that they can handle in order to deliver a valid recommendation. This becomes such systems as a good alternative to deal with information overload problems [11]. Most recommendation systems focus on the historical behaviour of users, since recommendations are generated by the similarity of searches of the users of other similar users, or by the rating that the user has given, in the past, to an item or an option. Indeed, to provide recommendations to a user, the system could consider the knowledge or experience of the same user. If the user does not have knowledge or experience, a valid recommendation can be provided in a category that the user has selected [12].

### C. Types of Recommendation Systems

There are different types of recommendation systems, each of which has its own approach about how to provide recommendations. Thus, these systems can be classified into:

*1) Recommenders based on collaborative filtering:* These systems focus on the items that received a rating from users [12]. This type of system is the most used since it helps to joint users with similar interests. This type of recommendation system does not need too much information about items, because the user is the one who really provides the information considered for the recommendation.

*2) Content-based recommenders:* These systems do not use the evaluation that the user provides about a product, but they use other parameters such as the information of the product itself or the user's profile [12]. This type of system is used more within scenarios where there are quite new products for which there is good information about the product and its characteristics.

*3) Knowledge-based recommenders:* In order to deliver a recommendation, these systems take all the available explicit knowledge about a product and a user, queries of the users, and information about what the expected result should look like [13]. The user can control the recommendations provided by different filters.

*4) Recommenders based on demographic information:* To make a recommendation, these systems consider user characteristics, such as: gender, age, education, etc. [14].

*5) Keyword based recommenders:* The operation of these systems is founded on measuring user preferences based on keywords. For this, an analysis of texts written by users is used to generate recommendations. In this type of recommendation system, users are classified as previous and active. From previous users, a set of keywords is extracted from their reviews and stored within a database. Thus, when an active user provides a new keyword and its weighted importance, the similarity of the keywords of the previous user's revisions with the new keyword is calculated [14].

*6) Hybrid recommenders:* These systems combine two or more types of the recommenders listed above. The aim of these systems is to provide better performance and improve recommendations [13].

Each of the recommendation systems mentioned has advantages and disadvantages, which are listed in Table I. For our research, it was essential that the recommendation system selected is one that involves the user as much as possible through ratings and that also takes into account the characteristics of the considered items (anomalies and vulnerabilities that would compromise computer security). For this reason, we used a hybrid system that combines both a recommender based on a collaborative filter and a recommender based on knowledge. The selected types and strategies are detailed below.

### D. Recommenders based on collaborative filtering

These systems are useful in environments where there is few content or knowledge associated with the elements. The recommendation is made to users who have relevant interests and preferences by calculating similarities between their profiles and behaviours [9]. Users create a group, which is called a neighbourhood, where a user gets recommendations for items that have been rated, or not, by other users in the same neighbourhood [9]. For its operation it is necessary that the users have read the same recommendation, which allows to group them as similar users (Fig. 1).

**Figura. I.5.** Página dos de artículo científico

TABLE I. ADVANTAGES AND DISADVANTAGES OF THE DIFFERENT TYPES OF RECOMMENDATION SYSTEMS [12] [14] [15]

| Advantages | Disadvantages |
|---|---|
| ***Recommenders based on collaborative filtering*** | |
| • No information about the products is needed<br>• Classifiers can be used to provide recommendations | • Cold start problem<br>• High cost to find the best neighbour<br>• "Black sheep" problem<br>• Data scarcity problem<br>• Scalability<br>• Quality |
| ***Content-based recommenders*** | |
| • Provides recommendations as soon as it has information about an available product | • Causality problem |
| ***Knowledge-based recommenders*** | |
| • Solve problems of collaborative filtering systems, such as: quality and cost of finding the best neighbour<br>• Improves causality problems<br>• It facilitates cold start | • Association rules between products and knowledge bases<br>• Complexity grows as the number of products grows |
| ***Recommenders based on demographic information*** | |
| • Best quality recommendation for the user | • Cold start problem |
| ***Keyword based recommenders*** | |
| • Can handle comments in the form of text reviews<br>• Can be integrated with social networks<br>• It can incorporate multi-criterion rating<br>• Good precision | • Difficulty for calculating similarities<br>• Keyword classification problem<br>• Weight calculation problem |
| ***Hybrid recommenders*** | |
| • Improved precision<br>• Improved performance<br>• It can overcome the problems of other recommendation systems | • Complex systems<br>• Expensive systems to implement |

The response provided by a recommendation system based on the collaborative filter may be one of two types: prediction and recommendation. Prediction is a rating of an item that would be given by a user. While the recommendation is the elements that the user likes or would like the most [9] (Fig. 2).

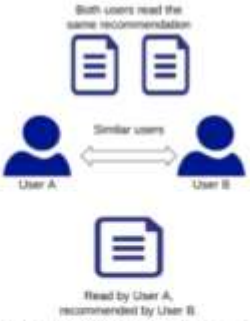



Fig. 1. Collaborative filtering working model.

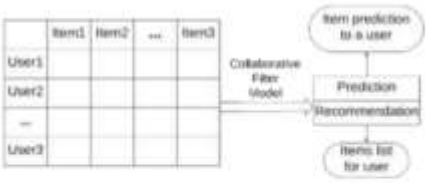



Fig. 2. Utility matrix.

To implement this recommender in our project, the neighbour-based strategy was used in order to make recommendations based on ratings from similar users [16].

*E. Knowledge-based recommenders.*

This recommender offers the possibility of exploiting the knowledge of a specific domain and thus provide expert recommendations to solve a problem [13].

For its operation it is necessary to create a database or knowledge base of preferred elements of the different users of the system (Fig. 3). In our case, the knowledge base was populated based on the knowledge of the domain defined in static information from security databases, and the contribution of experts' knowledge [13] as detailed in the following section.

## III. Research Methodology

For the implementation of the recommendation system, the steps outlined in the Fig. 4 were followed.

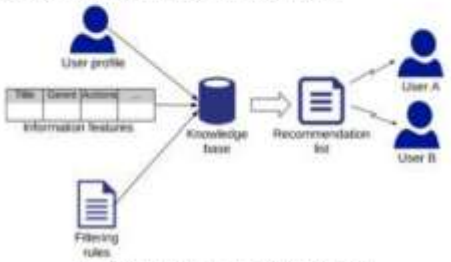



Fig. 3. Knowledge-based system operating model.


Fig. 4. Development process for the recommendation system.

**Figura. I.6.** Página tres de artículo científico

*1) Collection of relevant information:* First, we collected information from the websites of Symantec, OWASP, NIST the University of Trento-Italy and the CSIRT of the XXXX University. The rating of each anomaly proposed by the University of Trento-Italy was used to prepare the recommendations. We used web scraping techniques to scan the content of the web pages in order to obtain the information with which the recommendation system will run.

*2) Information processing:* After a process of verification and cleaning of the data obtained, the information was classified according to its level of criticality. For the classification of the information collected, the NumPy and Pandas libraries of Python were used. The name of the anomaly, a short description and its possible mitigation were identified. This information will be one input, in addition to the user's rating, to prepare recommendations.

*3) The implementation of the recommender system:* Once the information was verified, cleaned and classified, the recommendation system was implemented based on both a collaborative filter and knowledge. The recommender was designed so that once an attack to be mitigated is chosen, the system provides the recommendation with the best rating, and the five best alternative recommendations to the solution presented.

The system was developed applying an incremental iterative development model which consists of delivering functional prototypes. Python version 3 in an Anaconda environment was used at the core of the system. It was also used for the Web Scraping tool that gathering the data from different sources. For the visualization of the data, the workflow and the results, Jupyter Notebook and JupyterLab was used. GitHub repositories were used to control versions or prototypes.

*4) Development of a graphical interface:* To facilitate the use and to measure the effectiveness of the system, a graphical interface was developed. It allowed final users interact with the recommendation system and easily visualize the anomalies and recommendations suggested. The graphical interface was developed with Python. The Tkinter graphical library was selected for its clear, easy to program syntax and available documentation.

*5) Evaluation:* The recommendation system was evaluated by a group of experts in the area of cybersecurity and data analysis who carried out their evaluation using the Technology Acceptance Model (TAM) framework [17]. Thus, the evaluation was based on the two main dimensions of TAM: (1) the perceived utility defined as the subjective probability of a person that, by using a certain system, would improve their performance at work; and (2) the perceived ease of use that refers to the degree to which a person believes that using a certain system will be effortless [17].

## IV. System Description

### A. Logical Architecture

Our system is structured in two parts: a knowledge base and an anomaly ranking base. The knowledge base uses a flat file, where the information collected from the official security pages of Symantec, OWASP, NIST, the University of Trento-Italy and the CSIRT of the XXXX University are stored. This information consists of the attack identifier, the attack name, the criticality of the anomaly and the recommendation.

For the ratings, a second flat file is used. It stores the identifier of the user who made the recommendation rating and the identifier of the attack. The union of both files provides the necessary information for the operation of the recommendation system (Fig. 5).

For its operation, the system follows the following procedure (Fig. 6):

1. The knowledge base (File One) feeds the inference engine, which contains the rules that will be used to classify the collected information.
2. Once the anomalies have been ranked (File Two), the recommendation system, developed in Python, is fed. It will provide the recommendations for the incidents that the user is trying to solve.
3. For the system, the end users are also considered experts in the domain. Users benefit from existing recommendations in the system, however, as experts, they can contribute with their own recommendations, modify or remove existing recommendations. This will add feedback into the system, that will lead to improve future recommendations or adapt them to particular environments.

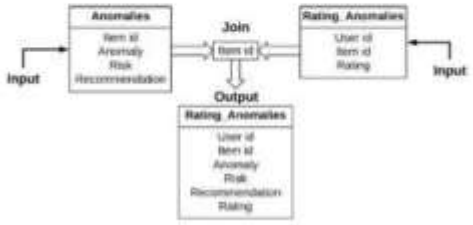



Fig. 5. Data structure

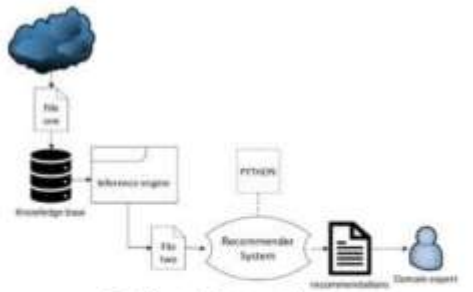



Fig. 6. Logical architecture of the system.

**Figura. I.7.** Página cuatro de artículo científico

*B. Physical Architecture of the recommendation system.*

The recommendation system is a local software that does not need any kind of installer and whose information is stored within the client computer that is running it. For collect external information, the system can connect to the internet to download and store data about anomalies in the database. Once these data is stored locally in the system, no more internet connection is required for its operation (Fig. 7).

*C. User interface.*

*System start screen:* On this screen the analyst will be able to enter the name of the anomaly for which he wishes to receive recommendations. From the content entered, the analyst will be able to carry out a search in a proposed list of the different anomalies registered (Fig. 8). The name of the anomaly is auto-completed as long as it is typed. If the anomaly is not registered, the system will display an informative message.

*Anomaly Results Screen:* On this screen the analyst will be able to see the five (5) anomalies closest (similar) to the one sought, whose recommendations could help mitigate the actual security issue (Fig. 9).


Fig 7. Physical architecture of the system.


Fig. 8. System anomalies list.

Fig. 9. Anomaly results screen.

*Recommendation Screen:* On this screen the analysts will be able to see the description and the recommendation(s) that would serve them to mitigate the anomaly selected on the results screen (Fig. 10).

## V. Evaluation

The recommendation system was evaluated by a group of experts in cybersecurity and data analysis. These experts were selected with the aim of cover a wide range of criteria in the use and utility of the recommendation system [18]. In this way, six experts who worked in academic and professional fields and belonging to public and private sectors were considered [18]. For the evaluation, we used the criteria established by the TAM model [17] with focus on the perceived ease of use and the perceived utility of the system. Also, some questions to understand the work of the experts were added.

The results found helped to understand the behaviour of each expert at the moment to provide a response to the anomalies that may arise in their daily work. This allowed to explore the way the recommender system can be useful to them. Thus, each of the experts react in different ways to an anomaly:

- Reporting the anomaly to the service desk.
- Try to resolve the anomaly the moment it occurs and seek help from partners depending on the severity and considering budget limits.
- Performing the vulnerability treatment process according to a CSIRT.
- Carrying out a vulnerability analysis, in which the presence of the vulnerability is confirmed and is scaled to the respective system administrator in order to take the necessary measures to resolve the vulnerability.
- Or, implementing a patch.

Almost all the experts confirmed that in their daily activities frequently face anomalies and vulnerabilities that can compromise computer security. Among these, the experts identified the following as the most recurrent: phishing, virus, malware, improperly configured services, SQL injection attacks, misleading window threats, Cross Site Scripting attacks, and obsolete applications' vulnerabilities.


Fig. 10. Description and recommendation for the anomaly.

**Figura. I.8.** Página cinco de artículo científico

Given these anomalies, and depending on its severity, the experts estimated that they can provide a response in a delay between one hour to one day. The extra time it takes to each expert to write a report is not taken into account.

Concerning our recommender, four of the six experts affirmed that the system satisfy their needs to provide a quick response to anomalies and vulnerabilities. The other two gave a positive evaluation, pointing out improvements that can be introduced such as implement the recommendation system as a web-based solution or include the source of the recommendations.

All six experts believe that thanks to the system they were able to improve the response time to anomalies and vulnerabilities that could compromise the computer security of their organizations.

## VI. Discussion

From our preliminary analysis, it was determined that the type of recommendation system that best suits the needs of both the user and the elements (in this case, the anomalies and vulnerabilities that compromise computer security) is a hybrid recommender based in: (1) a collaborative filter that allows the generation of a knowledge base, which is created jointly by the experts' judgements and information of security web sites; and (2) a knowledge-based recommender for dealing with the scarcity of the content or data that would be found in a recommender based only on a collaborative filter. This approach allowed us to provide recommendations qualified by expert judgement and, in this way, take advantage of both the knowledge and the experience of the computer security experts.

The expert judgement allowed to identify the worst vulnerabilities and anomalies within different entities. However, within each organization, the information about their worst vulnerabilities is confidential because the security of the organization could be very compromised if this information falls into the wrong hands. The fear of exposing this information constituted an important barrier that was mitigated by using initial information from the official pages of different entities (Symantec, OWASP, NIST, the University of Trento-Italy and the CSIRT of the XXXX University). As each of these sites provides different data sets, some tools were used to identify and get the information necessary to generate the knowledge base.

Despite the fact that some of the initial recommendations regarding anomalies and vulnerabilities did not adequately adjust to the reality of the organizations whose members evaluated the system, it is possible to affirm that the purpose of the system was fulfilled. Indeed, based on the results obtained, the recommendation system helped the experts to reduce the time they spent to solve a cybersecurity issue, it limited unnecessarily manual processes and reduced the subjectivity of the cybersecurity analyst. This is because having a tool that consolidates information about anomalies or vulnerabilities, with their respective recommendations, makes it easier not only to provide and respond to problems with a shorter response time, but also to prepare reports that serve to document them.

## VII. Conclusions

Recommendations based on collaborative filters help to generate the knowledge base with the help of expert judgement. In this way the experts can provide adequate recommendations about cybersecurity anomalies and vulnerabilities, and how to counter them. However, due to its main drawbacks with cold start issues (data scarcity and quality) the collaborative filter recommender was joined with a knowledge-based recommender to resolve the above-mentioned drawback. A recommendation system based on the content was discarded because they do not consider the assessment that a user can provide.

The datasets collected from universities and cybersecurity entities (i.e. Symantec, OWASP and NIST) allowed the elaboration of qualifications, the identification of the worst vulnerabilities and the best recommendations to counter them. However, it is data that could not always be used for every organization. In addition, the data present different nomenclature for anomalies and vulnerabilities, which can confuse end users. Thus, it is then necessary to include mechanisms that allow each organization to add its own information into the system or edit the existing one.

Finally, we conclude that the implementation of our recommendation system helped to improve the response time of cybersecurity analysts. It facilitated the execution of manual processes and reduced the subjectivity of the analyst. This was achieved thanks to the fact that the users can access to expert judgement, getting truthful information that facilitates a better decision to respond to a cyberattack.

## References


[1] Western Governors University, «What does a cyber security analyst do?,» 13 Agosto 2018. [En línea]. Available: https://www.wgu.edu/blog/what-does-cybersecurity-analyst-do1808.html. [Last access: 2 april 2019].

[2] A. Oltramari, N. Ben-Asher, L. Cranor, L. Bauer y N. Christin, «General Requirements of a Hybrid-Modeling Framework for Cyber Security,» 2014 IEEE Military Communications Conference, 20 November 2014.

[3] I. Herwono y F. A. El-Moussa, «A System for Detecting Targeted Cyber-Attacks Using Attack Patterns,» Springer International Publishing AG, nº 867, pp. 20-34, 9 June 2018.

[4] J. B. Schafer, J. A. Konstan y J. Riedl, «E-Commerce Recommendation Applications,» Data Mining and Knowledge Discovery, vol. 5, pp. 115-153, January 2001.

[5] R. Burke, «Hybrid Recommender Systems: Survey and Experiments,» User Modeling and User-Adapted Interaction, vol. 12, pp. 331-370, November 2002.

[6] CISCO, 2020. [En línea]. Available: https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html.

[7] E. Amoroso, Cyber Security, Primera ed., S. Press, Ed., New Jersey, 2006.

[8] Universidad Veracruz, 2020. [En línea]. Available: https://www.uv.mx/csirt/que-es-un-incidente-de-ciberseguridad. [Last access: January 2020].

[9] F. Isinkaye, Y. Folajimi y B. Ojokoh, «Recommendation systems: Principles, methods and evaluation,» Egyptian Informatics Journal, vol. XVI, nº 3, pp. 261-273, November 2015.

[10] T. Mahmood y F. Ricci, «Improving recommender systems with adaptive conversational strategies,» Proceedings of the 20th ACM conference on Hypertext and hypermedia, June 2009.


**Figura. I.9.** Página seis de artículo científico

[11] J. Shu, X. Shen, H. Liu, B. Yi y Z. Zhang , «A content-based recommendation algorithm for learning resources,» Multimedia Systems, vol. 24, p. 163–173, 2017 Mars 14.

[12] M. C. Martínez, «Sistemas de Recomendación basados en técnicas de predicción de enlaces para juegos en línea,» Madrid, 2017.

[13] H. Casanova, E. Ramos y H. Nuñez, «Sistema Basado en Conocimiento para Recomendación de Información Turística Venezolana,» III Simposio Científico y Tecnológico en Computación - SCTC 2014, May 2014.

[14] N. Vaidya y A. R. Khachane, «Recommender systems-the need of the ecommerce ERA,» 2017 International Conference on Computing Methodologies and Communication (ICCMC), July 2017.

[15] P. Pérez, «Recomendaciones en tiempo real mediante filtrado colaborativo incremental y real-time Big Data,» Madrid, 2016.

[16] A. Ruiz Iniesta, «Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos,» 2014.

[17] F. D. Davis, «Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,» MIS Quarterly, vol. XIII, nº 3, pp. 319-340, September 1989.

[18] Anonyzed.

**Figura. I.10.** Página siete de artículo científico

# ANEXO II. TABLAS RELACIONADAS AL PROCESO DE SCRUM

## II.1 EJECUCIÓN DE FASES SPRINT 0

**Tabla II.1:** Fechas de las fases del Sprint 0. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.2:** Backlog Sprint 0. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-1** | Como USUARIO AUTORIZADO, deseo recopilar información de las anomalías, mitigaciones y sus respectivos ratings, para poder observar que ataques se dan con más frecuencia y poder brindar una mejor recomendación. | Reuniones para definir información requerida con CSIRT | 1 |
| | | Reuniones para definir información requerida con CEDIA | 1 |
| | | Investigación de datasets de incidentes y vulnerabilidades en grupos de trabajo de distintas universidades del mundo | 2 |

**Tabla II.3:** Criterios de aceptación Sprint 0. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-1** | Obtener información de las entidades colaboradoras | Aprobado |
| | Tener la información de manera física y en la nube | Aprobado |

## II.2 EJECUCIÓN DE FASES SPRINT 1

**Tabla II.4:** Fechas de las fases del Sprint 1. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.5:** Backlog Sprint 1. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-2** | Como USUARIO AUTORIZADO, deseo analizar la información recopilada, para que los datos recopilados sean verídicos y no contengan información innecesaria. | Realizar un diagnóstico de las necesidades para quien va dirigido el sistema | 2 |
| | | Organizar la información para que pueda ser legible para el análisis | 2 |
| | | Seleccionar la información, la cual, satisfaga las necesidades encontradas | 2 |

**Tabla II.6:** Criterios de aceptación Sprint 1. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-2** | Tener claro las necesidades de los usuarios finales que van a utilizar el sistema | Aprobado |
| | Verificar que los datos tengan la suficiente documentación | Aprobado |

## II.3 EJECUCIÓN DE FASES SPRINT 2

**Tabla II.7:** Fechas de las fases del Sprint 2. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.8:** Backlog Sprint 2. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-3** | Como USUARIO AUTORIZADO, deseo clasificar la información previamente analizada y recopilada, para que la información este clasificada por su nivel de criticidad, además se pueda identificar claramente el nombre de la anomalía, una descripción corta y su posible mitigación. | Procesar la información obtenida del sprint anterior según la relevancia de esta | 1 |
| | | Identificar que la información contenga nombre de la anomalía y una descripción | 1 |
| | | Identificar incidentes y vulnerabilidades más críticas | 1 |
| | | Clasificar toda la información de incidentes y vulnerabilidades por su nivel de criticidad | 2 |

**Tabla II.9:** Criterios de aceptación Sprint 2. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-3** | Clasificar la información de manera legible | Aprobado |
| | Clasificar las anomalías / vulnerabilidades por su nivel de criticidad | Aprobado |

## II.4 EJECUCIÓN DE FASES SPRINT 3

**Tabla II.10:** Fechas de las fases del Sprint 3. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.11:** Backlog Sprint 3. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-4** | Como USUARIO AUTORIZADO, deseo investigar sobre los sistemas de recomendación, para que el sistema de recomendaciones se acople a las necesidades de los analistas. | Conocer las funcionalidades, las cuales brindan los sistemas recomendadores | 1 |
| | | Realizar una comparativa de ventajas y desventajas de los distintos sistemas recomendadores | 2 |
| | | Analizar el mejor sistema recomendador, el cual se ajuste de mejor manera a las necesidades de los usuarios finales | 1 |

**Tabla II.12:** Criterios de aceptación Sprint 3. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-4** | Escribir acerca de los sistemas recomendadores | Aprobado |
| | Investigar los distintos sistemas recomendadores | Aprobado |
| | Realizar un análisis de ventajas y desventajas de cada sistema recomendador | Aprobado |

## II.5 EJECUCIÓN DE FASES SPRINT 4

**Tabla II.13:** Fechas de las fases del Sprint 4. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.14:** Backlog Sprint 4. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-5** | Como USUARIO AUTORIZADO, deseo implementar un sistema de recomendación basado en rating, para que mediante el rating que se le ha ya otorgado se dará la recomendación. | Crear reglas de análisis de nivel de criticidad en las anomalías/vulnerabilidades | 1 |
| | | Programar en PHP la toma de datos para alimentar al sistema de recomendación | 2 |

**Tabla II.15:** Criterios de aceptación Sprint 4. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-5** | Escoger el mejor sistema recomendador, el cual se adapte a las necesidades de los usuarios | Aprobado |
| | Programar las reglas para el rating de las anomalías/vulnerabilidades | Aprobado |
| | Crear una consola para el manejo del sistema | Aprobado |

## II.6 EJECUCIÓN DE FASES SPRINT 5

**Tabla II.16:** Fechas de las fases del Sprint 5. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.17:** Backlog Sprint 5. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-6** | Como USUARIO AUTORIZADO, deseo ingresar el nombre de la anomalía, por la cual se va a generar la recomendación, para que el analista pueda ingresar libremente el ataque que él desea mitigar sin ninguna restricción. | Verificar el ingreso de datos al sistema | 1 |
| | | Ingresar la anomalía/vulnerabilidad en la consola del sistema | 1 |
| | | Corregir errores en lectura de datos al ingresar por consola la anomalía/vulnerabilidad que se desea buscar | 2 |
| **UH-CS-7** | Como USUARIO AUTORIZADO, deseo visualizar las 5 mejores recomendaciones para dicha anomalía, la cual incluirá la anomalía ingresa da previamente, para que el analista escoja la | Verificar que las recomendaciones, las cuales se visualizan sean las correctas para la anomalía/vulnerabilidad buscada | 2 |
| | | Visualizar las mejores | 1 |

| | | | |
|---|---|---|---|
| | mejor recomendación basada en rating de otros usuarios. | recomendaciones para la anomalía/vulnerabilidad buscada | |

**Tabla II.18:** Criterios de aceptación Sprint 5. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-6** | Al ingresar la anomalía/vulnerabilidad el sistema debe tomar la entrada y comparar con la BDD de conocimiento | Aprobado |
| | Al encontrar problemas con la lectura de la entrada de datos al sistema, identificar el error y darle las correcciones necesarias | Aprobado |
| **UH-CS-7** | Al ingresar la anomalía/vulnerabilidad el sistema presentará las principales recomendaciones con el mejor rating dado por los usuarios | Aprobado |

## II.7 EJECUCIÓN DE FASES SPRINT 6

**Tabla II.19:** Fechas de las fases del Sprint 6. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.20:** Backlog Sprint 6. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-8** | Como USUARIO AUTORIZADO, deseo desarrollar una interfaz gráfica, para que el analista pueda visualizar de mejor manera tanto las anomalías como sus respectivas recomendaciones. | Desarrollar pantalla de inicio para el ingreso de anomalías | 1 |
| | | Desarrollar pantalla donde se visualice anomalías similares a la ingresada | 1 |
| | | Desarrollar pantalla la cual contenga las recomendaciones de la anomalía seleccionada | 2 |

**Tabla II.21:** Criterios de aceptación Sprint 6. Elaborado por autores.

| ID Historia | Criterio de aceptación | Aprobado o Rechazado |
|---|---|---|
| **UH-CS-8** | Ingresar anomalías las cuales permitan buscar dentro de la base de conocimiento | Aprobado |
| | Permitir seleccionar e interactuar con anomalías registradas | Aprobado |
| | Poder visualizar las recomendaciones adecuadas con su respectiva anomalía | Aprobado |

## II.8 EJECUCIÓN DE FASES SPRINT 7

**Tabla II.22:** Fechas de las fases del Sprint 7. Elaborado por autores.

| Fecha | Fase |
|---|---|
| **verificar** | Sprint Planning<br>• Sprint Goal<br>• Elaboración del Sprint Backlog |
| **verificar** | Ejecución del Sprint<br>• Daily Scrum<br>• Ejecución de las tareas<br>• Trabajo pendiente (Burndown chart) |
| **verificar** | Inspección y Adaptación<br>• Sprint Review<br>• Sprint Retrospective<br>• Actualización y refinamiento del Product Backlog y Release Planning |

**Tabla II.23:** Backlog Sprint 7. Elaborado por autores.

| ID Historia | Historia de Usuario | Tarea | Esfuerzo Estimado |
|---|---|---|---|
| **UH-CS-9** | Como USUARIO AUTORIZADO, deseo encuestar expertos, para que el analista pueda brindar comentarios de uso del sistema de recomendación. | Realizar encuestas a los expertos a quienes se les presente el sistema de recomendación | 1 |
| | | Clasificar las respuestas recibidas por los encuestados | 2 |

**Tabla II.24:** Criterios de aceptación Sprint 8. Elaborado por autores.

| ID Historia | Criterio de aceptación | Estado |
|---|---|---|
| **UH-CS-9** | Obtener respuestas acerca del funcionamiento y usabilidad del sistema | Aprobado |
| | Generar informe de resultados acordes a las encuestas realizadas | Aprobado |