UNIVERSIDAD DE TARAPACÁ

FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática

Proyecto IV

"Sistema de gestión de seguridad en un ambiente de servidores"

Security Control Center

Integrantes: Scarlet Gavia Mondaca
Empresa o Unidad: DICI
Profesor: Diego Aracena Pizarro
Asignatura: Proyecto IV

Arica, Chile
2025

Índice

Índice de figuras

Índice de tablas

1. Introducción

La ciberseguridad es una rama muy importante en la actualidad, a través de ella podemos proteger nuestros datos e información importante, ya que garantiza la protección, respaldo y resguardo de los datos, por ello, es esencial que los estudiantes adquieran habilidades para defender sus dispositivos y detectar posibles ataques de agentes externos.

Este proyecto tiene como objetivo diseñar e implementar un sistema de gestión de ciberseguridad informática, para llevar a cabo este proyecto se comenzará por realizar las instalaciones de distintos sistemas operativos de distribución “Linux” en cada uno de los servidores, los cuales tendrán distintas herramientas para que los alumnos puedan aplicar ataques y defensas controladas, se liberaran los puertos señalados por el cliente, se llevará un catastro de ruta de cada proceso que se realiza semanalmente, esto se llevará a cabo con un dashboard para llevar un monitoreo de los 4 servidores en simultáneo, este dashboard permitirá reiniciar los servidores y ejecutar cualquier aplicación que el cliente quiera instalar.

Asimismo, luego de estas configuraciones, el sistema deberá ser capaz de monitorear, registrar y analizar cada una de las actividades realizadas, ya sea en el servidor físico como ataques externos, con estos datos registrados en el sistema, este debe ser capaz de realizar un análisis, identificando y registrando cada intento de intrusión, lo que permitirá al cliente ver los ataque y poder realizar un análisis de estas capturas.

2. Descripción de la empresa

El Departamento de Ingeniería en Computación e Informática ha trabajado conscientemente en el proceso de transformación de estudiantes en Ingenieros y como tales deben ser capaces de analizar y sistematizar la información, con el fin de alcanzar los objetivos organizacionales de la empresa, tanto nacional como internacional, mediante el uso de sistemas computacionales distribuidos. Darles una formación, para que sean capaces de diseñar, desarrollar e implantar sistemas para administrar información útil en la toma de decisiones usando equipo computacional, a la vez que utilice metodologías y facilidades para el desarrollo general de sistemas complejos de software base y de sistemas en particular, generando tecnología nacional.

EL DICI tiene como misión crear, difundir y hacer uso de las tecnologías de información y comunicación en beneficio de la sociedad y la formación de ingenieros con sólidas habilidades técnicas y sociales. Esta misión involucra un compromiso y una responsabilidad con la región, el país, la Universidad de Tarapacá y el DICI, ya que estos principios y valores, son los mismos que establece nuestra casa de estudios superiores.

Cuando se habla de crear, difundir y hacer uso de las tecnologías de información y comunicación en beneficio de la sociedad se quiere expresar como una unidad académica que centra su quehacer no sólo en la formación de los profesionales en el área que le compete, sino, además, en la investigación, la extensión, la formación de recursos humanos (postgrado, postítulo) y en la prestación de servicios que la región requiere.

Figura 1: Logo de la empresa.

3. Definición del proyecto

3.1. Contexto

La computación y la tecnología son parte de nuestras vidas, estudiar computación te permite formar parte del grupo de personas que desarrollan los sistemas que están dando forma a los estilos de vida del futuro, la práctica de la ciberseguridad es esencial para esta rama de estudios, ya que permite que los alumnos desarrollen habilidades en defensa de sistemas y detección de ataques, sin embargo, es necesario que estas prácticas se realicen en un entorno controlado, donde los riesgos sean mínimos y la actividad de los alumnos pueda ser supervisada adecuadamente.

3.2. Problema

Los estudiantes aplicaran ataques y defensas en sistemas reales para desarrollar habilidades efectivas, sin embargo, estas prácticas presentan riesgos si se realizan en entornos no controlados, ya que los servidores podrían dañarse o los datos podrían comprometerse, el laboratorio carece de herramientas que les permitan supervisar y registrar las actividades de estos servidores, lo que dificulta al cliente ver las actividades realizadas.

3.3. Solución

Para dar solución a esta problemática, se propone la implementación de un sistema de monitoreo para un laboratorio de ciberseguridad controlado, este sistema permitirá al cliente monitorear, registrar y analizar las actividades realizadas en los servidores, alertando sobre ataques y permitiendo realizar defensa.

4. Objetivos

4.1. Objetivo General

Desarrollar un sistema de monitoreo que permita visualizar métricas, logs y alertar al administrador sobre las actividades sospechosas realizadas en los servidores.

4.2. Objetivos Específicos

- Ver el estado de los servidores.
- Estudiar distintas herramientas para la captura de datos de los servidores.
- Desarrollar el sistema en un servidor central para comenzar la captura de los datos.
- Mostrar métricas de CPU, RAM y sistema operativo.
- Instalar herramientas seleccionadas en los servidores.
- Analizar logs históricos con gráficos.
- Representar tácticas y técnicas.
- Realizar alertas en caso de actividad sospechosa.

5. Planificación del proyecto

5.1. Metodología ágil de prototipo descartable

Se utilizará una la metodología ágil “prototipo descartable”, ya que esta permite explorara, aclarar y validar requerimientos antes de construir la solución final, esta consiste en crear una versión preliminar, simplificada y parcialmente funcional del producto, el objetivo principal es facilitar la comunicación con el cliente, detectar errores, comprender necesidades reales y reducir la incertidumbre durante las etapas iniciales del proyecto.

Este prototipo no está destinado a evolucionar ni a formar parte del producto final, sino que se elimina una vez que se haya cumplido su función de análisis, validación y retroalimentación.

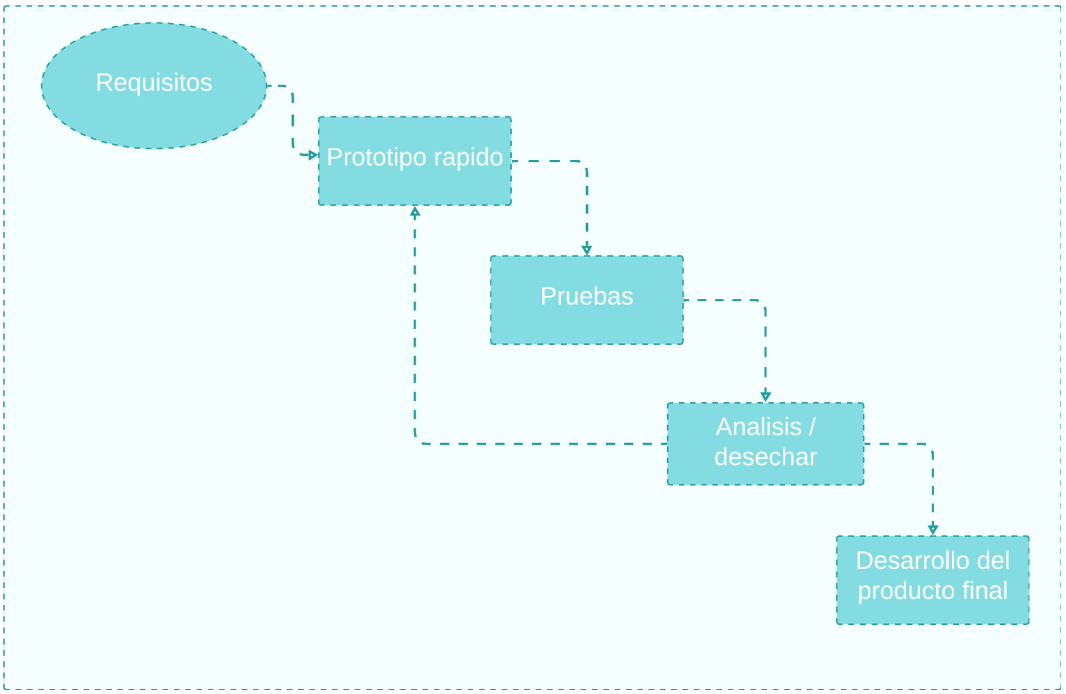

Figura 2: Metodología.

6. Carta Gantt

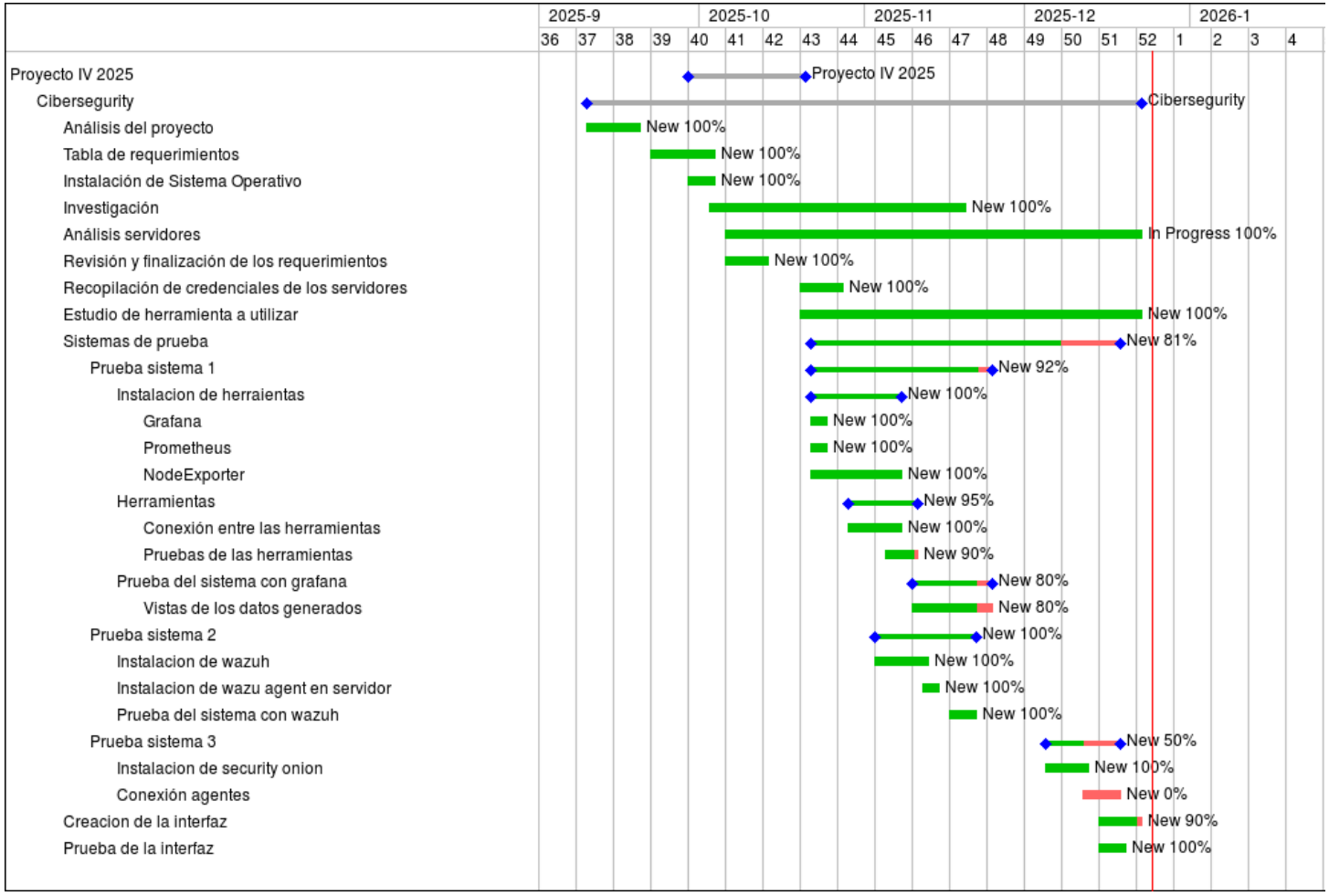

Figura 3: Carta Gantt.

7. Requisitos del sistema

7.1. Requisitos de Alto Nivel

ID	*Definición del requisito*
RAN1	Facilitar la supervisión y el análisis de las actividades realizadas.
RAN2	Garantizar servidores protegidos.
RAN3	El sistema debe contar con la tríada de ciberseguridad, pentesting, peritaje informáticos y seguridad informática en criptografía.

Tabla 1: Requisitos de alto nivel

7.2. Requisitos funcionales

ID	*Definición del requisito*	*Importancia*
RF1	El sistema debe tener 2 perfiles de acceso.	**muy alto**
RF2	El sistema debe permitir que el perfil de administrador modifique información de los servidores.	**muy alto**
RF3	El sistema debe permitir que el perfil de usuario solo pueda acceder a los datos registrados, sin opción de modificar.	**muy alto**
RF4	El sistema debe registrar todas las acciones y ataques realizados.	**alto**
RF5	El sistema debe mostrar los ataques y las vulnerabilidades detectadas.	**alto**
RF6	El sistema debe permitir la gestión de múltiples servidores.	**alto**
RF7	El sistema debe permitir un monitoreo y seguimiento de los ataques ejecutados.	**alto**
RF8	El sistema debe mostrar en pantalla el tiempo real de los eventos.	**medio**
RF9	El sistema debe mostrar los procesos de los servidores.	**medio**
RF10	El sistema debe registrar cualquier actividad realizada en los servidores (reinicio, apagado,etc).	**medio**

Tabla 2: Requisitos funcionales

7.3. Requisitos no funcionales

ID	*Definición del requisito*
RNF1	El sistema debe asegurar que los ataques realizados no afecten a otros sistemas de la empresa.
RNF2	Los servidores y el sistema deben estar siempre disponibles.
RNF3	El sistema debe ser capaz de manejar múltiples conexiones y acciones simultáneas.
RNF4	El sistema debe tener una interfaz intuitiva y permitir acceder fácilmente a la información de cada servidor.
RNF5	El sistema debe permitir agregar más servidores sin necesidad de reestructuración mayor.
RNF6	El sistema debe ser compatible con dashboard para la captura de datos.

Tabla 3: Requisitos no funcionales.

8. Análisis del sistema

8.1. Modelo de contexto

En el siguiente diagrama se explica el funcionamiento del sistema, en el cual se puede ver que los hackers o intrusos atacan los servidores, luego el sistema reconoce los ataques, analiza el servidor y finalmente el sistema alerta al administrador los ataques realizados, mientras que el usuario básico sólo puede observar las estadísticas de lo sucedido.

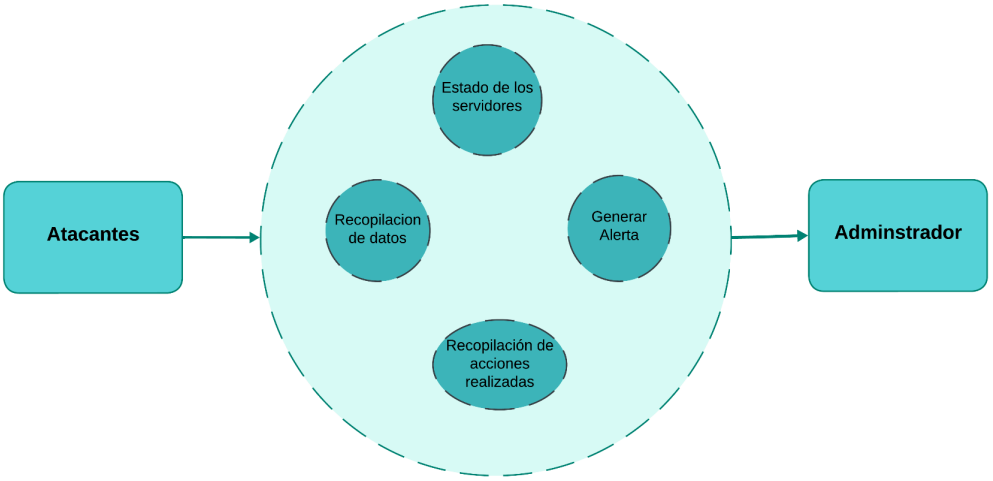

Figura 4: Modelo de contexto.

8.2. Casos de Uso del sistema

A partir del modelo de contexto presentado en la Figura 4, se identifican los principales casos de uso del sistema, los cuales representan las funcionalidades observables por el administrador, cada caso de uso se asocia a uno o más subsistemas.
En la siguiente tabla se presenta un resumen de los subsistemas del modelo contexto y los casos de uso identificados.

ID	Subsistema	Casos de uso asociados
CU-01	Estado de los servidores	Visualizar estado de los servidores
CU-02	Recopilación de datos	Visualizar métricas del agente.
CU-03	Generación de alertas	Consultar alertas de seguridad.
CU-04	Recopilación de acciones realizadas	Consultar logs históricos, filtrar logs, Visualizar gráfico de eventos ,analizar MITRE ATT&CK.

Tabla 4: Subsistemas.

8.3. Diagrama de casos de Uso

El diagrama de casos de uso representa las interacciones entre el administrador y el sistema, en este diagrama se muestran las funcionalidades principales que el sistema ofrece al usuario destacando aquellas relacionadas con la recopilación y análisis de acciones realizadas, debido a su mayor complejidad funcional.

En el diagrama de casos de uso se definieron los actores Usuario y Administrador, donde el Administrador hereda las funcionalidades del Usuario y posee privilegios adicionales, tales como la administración de usuarios del sistema.

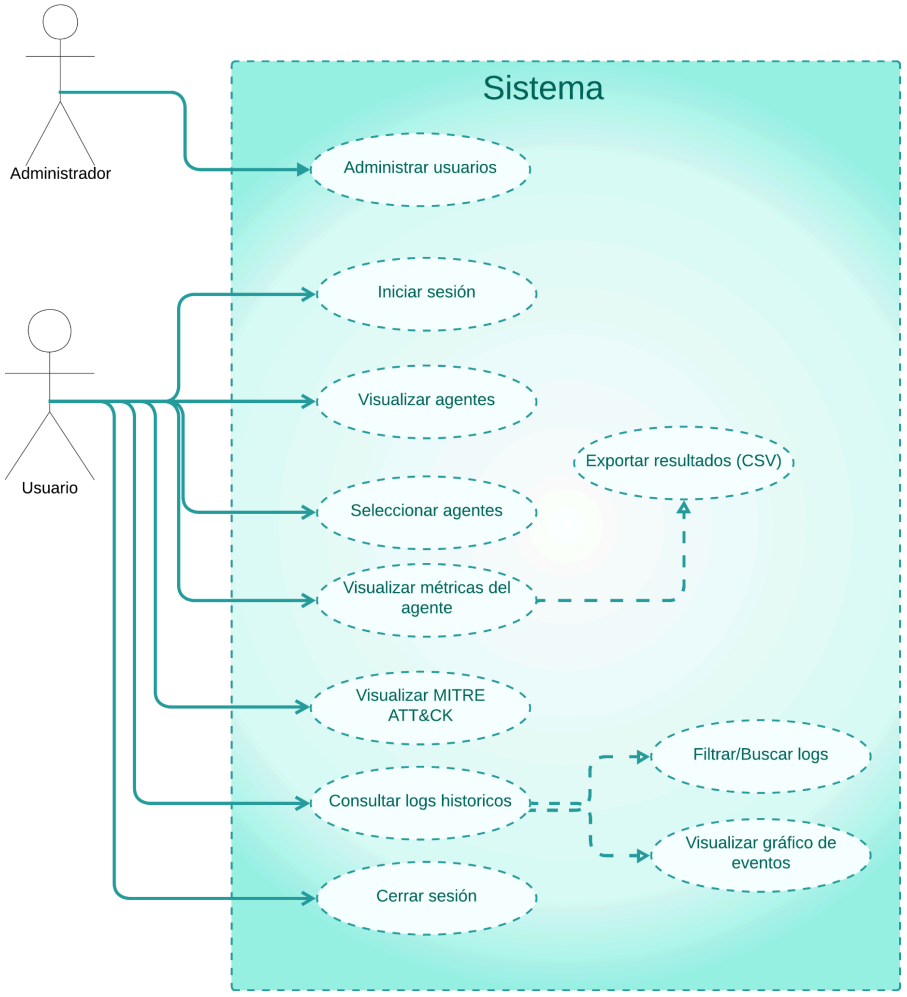

Figura 5: Diagrama de casos de usos

8.4. Análisis de casos de uso principales

CU–01	**Visualizar estado de los servidores**
Actor	Administrador
Descripción	El administrador puede visualizar el estado general de los servidores monitoreados, incluyendo información básica como disponibilidad, sistema operativo y estado de conexión.
Flujo principal	1.- El administrador accede al módulo de agentes 2.-El sistema muestra la lista de servidores y su estado actual

Tabla 5: CU-01.

CU–02	**Visualizar métricas del agente**
Actor	Administrador
Descripción	El sistema permite visualizar métricas asociadas a un agente seleccionado, tales como uso de CPU, memoria RAM y datos del sistema operativo.
Flujo principal	1.- El administrador selecciona un agente. 2.- El sistema recopila las métricas disponibles. 3.- El sistema presenta la información en forma gráfica y textual.

Tabla 6: CU-02.

CU–03	**Consultar logs históricos**
Actor	Administrador
Descripción	El administrador puede consultar los registros históricos de eventos generados por los agentes, permitiendo analizar acciones y eventos de seguridad ocurridos en un período determinado.
Flujo principal	1.- El administrador accede al módulo de logs históricos. 2.- El sistema recupera los eventos asociados al agente seleccionado. 3.- El sistema muestra el listado de logs

Tabla 7: CU–03

CU–04	Filtrar y buscar logs
Actor	Administrador
Descripción	El sistema permite filtrar los registros históricos por nivel de severidad y realizar búsquedas textuales, facilitando el análisis de eventos relevantes.
Flujo principal	1.- El administrador selecciona un nivel o ingresa un texto de búsqueda. 2.- El sistema filtra los logs disponibles. 3.- El sistema actualiza la visualización de resultados.

Tabla 8: CU-04

CU–05	Visualizar gráfico de eventos
Actor	Administrador
Descripción	El sistema presenta una visualización gráfica de los eventos registrados, permitiendo observar tendencias y frecuencias de eventos a lo largo del tiempo.
Flujo principal	1.- El administrador accede a la vista de gráficos. 2.- El sistema procesa los eventos históricos. 3.- El sistema genera el gráfico correspondiente.

Tabla 9: CU-05

CU–06	Analizar técnicas MITRE ATT&CK
Actor	Administrador
Descripción	El administrador puede visualizar las técnicas y tácticas MITRE ATT&CK asociadas a los eventos detectados, facilitando el análisis de patrones de ataque y comportamientos maliciosos.
Flujo principal	1.- El administrador accede al módulo MITRE. 2.- El sistema presenta la matriz MITRE con las técnicas detectadas

Tabla 10: CU-06

8.5. Modelo BPMN

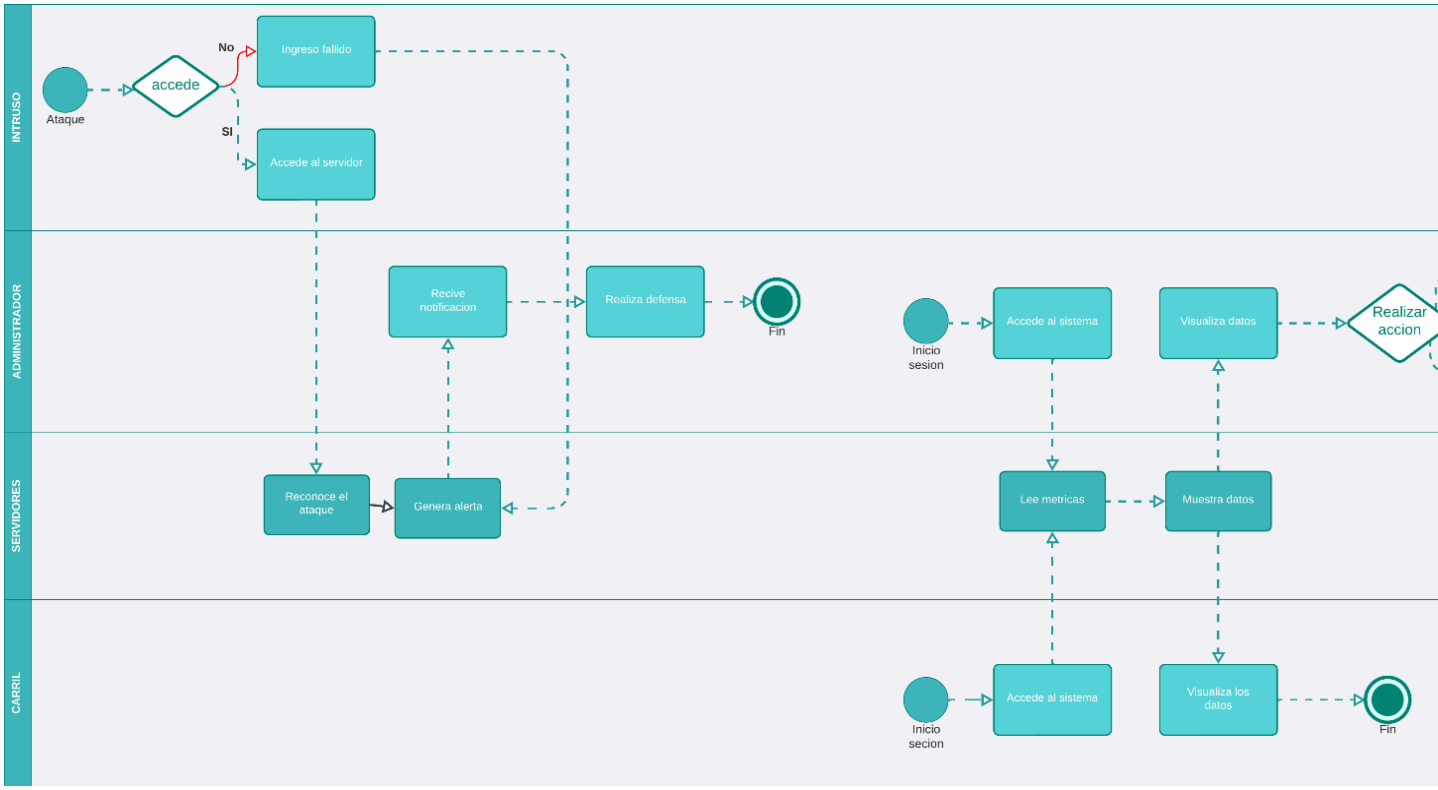

Figura 6: Modelo BPMN

9. Herramientas a utilizar

Se utilizaron distintas herramientas para el análisis de datos y alerta de ataques, para la primera prueba se utilizó:

	Nombre	Descripción
1	Grafana	Plataforma de código abierto que permite crear paneles de control interactivos y dinámicos que muestran métricas de rendimiento en tiempo real a través de gráficos.
2	Prometheus	Sistema de código abierto que permite recopilar y almacenar métricas de los sistemas objetivos, proporciona el lenguaje de consulta PromQL para el análisis de datos.
3	Alertmanager	Herramienta de código abierto que gestiona las alertas generadas por sistemas como Prometheus.
4	Node Exporter	Es un componente de la plataforma prometheus, recopilar métricas del sistema operativo y hardware de un servidor.

Tabla 11: Herramientas a utilizar 1ra prueba.

Para la segunda prueba se utiliza la siguiente herramienta:

	Nombre	Descripción
1	Wazuh	Plataforma para detectar amenazas en servidores, auditar cumplimiento, centralizar alertas y servir como capa de visibilidad en infraestructuras físicas, virtuales, en la nube y en contenedores.
2	Elasticsearch	Motor distribuido de búsqueda de logs, métricas, eventos, almacenamiento y cualquier dato indexable.
3	Kibana	Interfaz web de visualización y administración para Elasticsearch, crear dashboards, explorar datos con Discover, gestionar índices, visualizar mapas, crear visualizaciones y usar plugins
4	Filebeat	Recopila archivos de log, lee nuevos eventos en tiempo real, aplica procesadores básicos, y envía a destino configurado.

Tabla 12: Herramientas a utilizar 2da prueba.

Para la vista del cliente se utilizarán:

	Nombre	Descripción
1	Angular	Framework de desarrollo front-end basado en TypeScript, diseñado para crear aplicaciones web dinámicas, modulares y de alto rendimiento.
2	Node.js	Entorno de ejecución de JavaScript del lado del servidor, basado en el motor V8 de google, permite ejecutar código JavaScript fuera del navegador.
3	Bootstrap	Framework de diseño css orientado a la creación de interfaces web .
4	Visual Studio Code	Editor de código ligero, multiplataforma y altamente extensible, ofrece funcionalidades avanzadas como autocompletado inteligente, depuración integrada.

Tabla 13: Herramientas a utilizar framework.

10. Alcance del proyecto

El proyecto consiste en el desarrollo e implementación de un sistema de monitoreo y gestión de servidores, diseñado para mejorar la visibilidad, seguridad y control operativo de una infraestructura compuesta por un servidor central para el monitoreo y otros servidores los cuales serán monitorizados.
El sistema utiliza Wazuh como plataforma de monitoreo y detección de intrusiones, complementado con una interfaz web personalizada desarrollada con Angular y Bootstrap, que permitirá a los usuarios visualizar y administrar la información de manera centralizada.

10.1. Elementos incluidos dentro del alcance

- Implementación de Wazuh Manager
- Instalación y configuración del servidor Wazuh.
- Conexión y registro de los servidores adicionales mediante Wazuh Agent.
- Configuración de políticas básicas de monitoreo, alertas e integridad de archivos.
- Monitoreo en tiempo real

10.2. Visualización

El sistema permitirá que el usuario final pueda ver:

- Intentos de intrusión (eventos de seguridad).
- Actividades anómalas dentro de los servidores.
- Logs del sistema y comportamiento de procesos.
- Panel unificado para ver los tres servidores juntos o individualmente.
- Estado general de los servidores.
- Alertas críticas.
- Recursos del sistema (CPU, RAM, disco, red).
- Información de seguridad y eventos recientes.
- Gestión remota de los servidores (según permisos)

10.3. Funcionalidades que tiene el sistema

- Reiniciar servidores.
- Actualizar paquetes o servicios.
- Ejecución de acciones controladas con autorización del usuario.
- Gestión de usuarios
- Creación de nuevos usuarios dentro del sistema web.

Estas acciones solo son permitidas para el usuario con rol de administrador.

10.4. Usuarios permitidos por el sistema

- Administrador: Usuario con acceso completo a configuración y acciones sobre los servidores.
- Usuario estándar: Usuario con acceso solo a visualización del estado de los servidores.
- Control y auditoría de acciones realizadas.

11. Prototipo

Para la pantalla principal se presentará un formulario de inicio de sesión, en el cual según el nombre el sistema abrirá el usuario correspondiente, ya sea de administrador o de usuario.

Figura 7: Pantalla inicial.

Luego en la segunda pantalla se mostrará una pestaña con opciones como inicio, la cual nos llevará a una página de inicio y a la página de los paneles de control, en el final de esta pestaña se puede ver la opción de configuraciones.

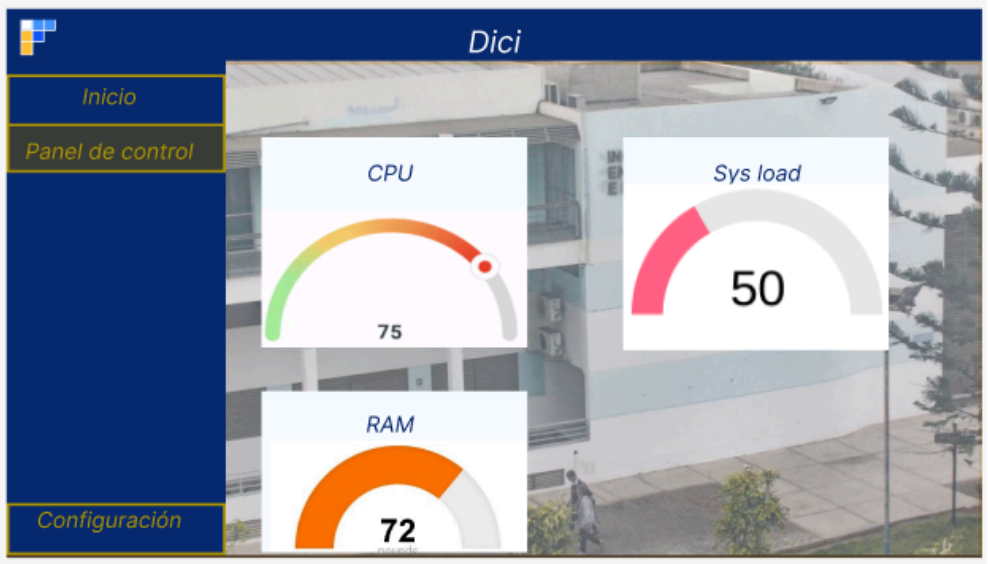

Figura 8: Panel de control.

12. Arquitectura del sistema

La arquitectura del sistema representada en la Figura 9, se organiza en tres capas principales: capa de aplicación, capa de gestión y capa de agentes.

- **Capa de aplicación:** El administrador accede al sistema a través de una aplicación web desarrollada con “Angular”, apoyada en “Bootstrap” para la interfaz gráfica y “Node.js” como capa intermedia de servicios. Esta capa de aplicación permite la visualización y gestión de la información de seguridad mediante un dashboard interactivo.
- **Capa de gestión:** Está compuesta por los componentes principales de “Wazuh”, incluyendo “Wazuh Manager”, “Wazuh API REST”, “Wazuh Ruleset” y el “motor de análisis”, los cuales se encargan de procesar los eventos de seguridad generados por los agentes, los resultados de este análisis son almacenados en “Elasticsearch”, que actúa como el “almacén de datos históricos”, permitiendo conservar logs, alertas y eventos para su posterior consulta y análisis.
- **Capa de agentes:** Está formada por “Wazuh Agents”, los cuales se instalan en los servidores a monitorear, estos agentes recopilan información del sistema y eventos de seguridad, enviándolos al Wazuh Manager para su procesamiento.

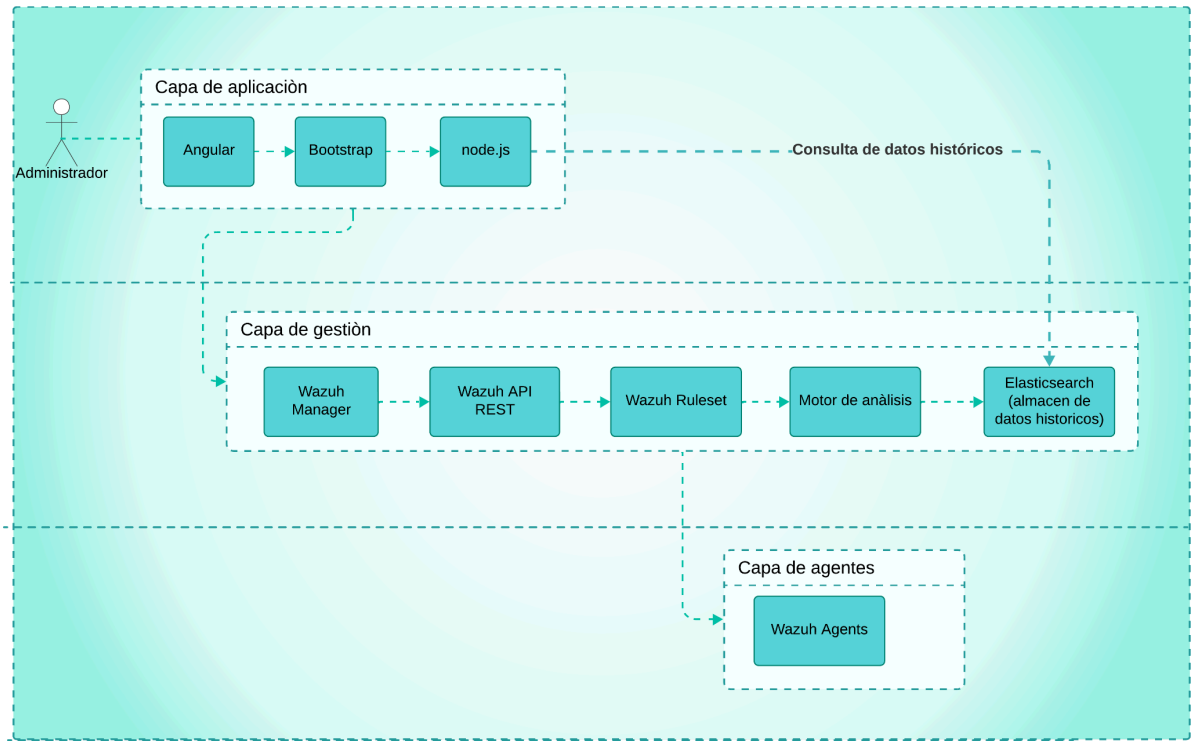

Figura 9: Arquitectura del sistema.

13. Conclusión

En el presente informe se abordaron los procesos de organización y diseño del “Sistema de gestión de seguridad en un ambiente de servidores”, cuyo objetivo principal es proporcionar al administrador una herramienta que permita monitorear múltiples servidores en tiempo real, optimizando así la gestión y seguridad de la infraestructura, este sistema cuenta con un dashboard centralizado, el cual facilita la visualización de métricas, eventos y logs históricos, además de un mecanismo de alertas que permite detectar posibles intrusiones o incidentes de manera oportuna.

Para el desarrollo de este proyecto, se realizó un análisis de distintas herramientas y tecnologías, se elaboró un prototipo funcional y se diseñaron diversos diagramas los cuales permitieron una mejor comprensión y planificación de la solución.

En conjunto, estos elementos contribuyeron a definir una base sólida y estructurada, que facilita futuras implementaciones, mejoras y escalabilidad del sistema, permitiendo su adaptación a distintos entornos de servidores y necesidades de seguridad.

14. Referencias

1. Grafana Labs. (s. f.). *Grafana: The open and composable observability platform | Grafana Labs*. https://grafana.com/
2. *Getting started | Prometheus*. (s. f.). https://prometheus.io/docs/prometheus/latest/getting_started/
3. Wazuh. (s. f.). *Quickstart · Wazuh documentation*. https://documentation.wazuh.com/current/quickstart.html
4. *Lucidchart | Diagramas creados con inteligencia*. (s. f.). Lucidchart. https://www.lucidchart.com/pages/es
5. Contributors, M. o. J. T. A. B. (s. f.). *Get started with Bootstrap*. https://getbootstrap.com/docs/5.3/getting-started/introduction/

15. Anexos

1) ScarletGavia. (s. f.). *GitHub - ScarletGavia/wazuh-monitor*. GitHub. https://github.com/ScarletGavia/wazuh-monitor