
Especificación de requisitos de software

Proyecto: Plataforma Web de Análisis y
Monitoreo del uso de recursos del sistema
operativo Linux



Ficha del documento

Fecha	Revisión	Autor	Modificación
6/3/2025	V-100	Equipo “PEM”	6/3/2025



Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	5
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	7
1.6 Resumen	8
2 DESCRIPCIÓN GENERAL	8
2.1 Perspectiva del producto	8
2.2 Funcionalidad del producto	8
2.3 Características de los usuarios	8
2.4 Restricciones	9
2.5 Suposiciones y dependencias	10
2.6 Evolución previsible del sistema	10
3 REQUISITOS ESPECÍFICOS	10
3.1 Requisitos comunes de los interfaces	10
3.1.1 Interfaces de usuario	10
3.1.2 Interfaces de hardware	10
3.1.3 Interfaces de software	11
3.1.4 Interfaces de comunicación	11
3.2 Requisitos funcionales	11
3.2.1 RF-01: Monitoreo de CPU	11
3.2.2 RF-02: Monitoreo de Memoria RAM	11
3.2.3 RF-03: Monitoreo de Red	12



Especificación de requisitos de software

3.2.4	RF-04: Monitoreo de Procesos	12
3.2.5	RF-05: Monitoreo de Disco Duro	13
3.2.6	RF-06: Dashboard	13
3.2.7	RF-07: Gráficos en tiempo real	14
3.2.8	RF-08: Sistema de Alertas	14
3.2.9	RF-09: Autenticación de Usuarios	14
3.2.10	RF-10: Configuración de Umbrales	15
3.2.11	RF-11: Almacenamiento de Datos	15
3.2.12	RF-12: Interfaz de Usuario	16
3.3	Requisitos no funcionales	16
3.3.1	RNF-01: Requisitos de rendimiento	16
3.3.2	RNF-02: Seguridad	16
3.3.3	RNF-03: Fiabilidad	16
3.3.4	RNF-04: Disponibilidad	16
3.3.5	RNF-05: Mantenibilidad	17
3.3.6	RNF-06: Portabilidad	17
3.4	Otros requisitos	17
4	APÉNDICES	17
4.1	GitHub del Sistema	17
4.2	Mockups del Sistema	17
4.3	Diagrama Entidad Relación	17
4.4	Diagrama de caso de uso	17



1 Introducción

El desarrollo de software consta de varias fases: planificación, análisis, desarrollo pruebas y mantenimiento todo compone un ciclo iterativo, para cualquier proceso que se necesite automatizar, para ello se realiza un proceso de documentación donde va dirigido a stakeholders, incluyendo el equipo de desarrollo. Se ha estructurado con los estándares de la IEEE 830-1998 para la especificación de los requisitos de software, proporcionando una descripción clara de la arquitectura del sistema.

1.1 Propósito

Este documento busca establecer los requisitos funcionales y no funcionales del sistema considerando las necesidades a requerir, utilizando un lenguaje natural para una máxima comprensión de este mismo, así asegura un proceso correcto para el desarrollo de una plataforma de monitoreo de recursos del sistema operativo Linux.

1.2 Alcance

Este documento tiene como objetivo proporcionar un sistema de monitoreo en tiempo real que permita la visualización de métricas de recursos del sistema operativo, análisis de rendimiento mediante gráficos interactivos, sistema de alertas para situaciones críticas y proporcionar un dashboard de los recursos utilizados en un sistema operativo Linux. El sistema se implementará utilizando Python como lenguaje principal y el framework Django para la arquitectura, proporcionando capacidades de monitoreo en tiempo real de las métricas críticas del sistema.

1.3 Personal involucrado

Tabla 1 Líder de equipo

Nombre	Elkin Andrés Pabón Gonzales
Rol	Líder de equipo
Categoría profesional	Estudiante de Ingeniería en software
Responsabilidades	Encargado de la creación del documento de ERS cumpliendo las expectativas del sistema solicitado y también las normas IEE830
Información de contacto	Numero de contacto: 0995473517 Correo: eapabon@espe.edu.ec
Aprobación	

Tabla 2 Equipo de ERS (Josue Marin)

Nombre	Josue Isaac Marin Alquina
Rol	Equipo de Documentación de ERS
Categoría profesional	Estudiante de Ingeniería en software



Especificación de requisitos de software

Responsabilidades	Encargado de la creación del documento de ERS cumpliendo las expectativas del sistema solicitado y también las normas IEE830
Información de contacto	Numero de contacto: 0991267695 Correo: jimarin@espe.edu.ec
Aprobación	

Tabla 3 Equipo de ERS (Isaac Escobar)

Nombre	Isaac Alejandro Escobar Rivadeneira
Rol	Equipo de Documentación de ERS
Categoría profesional	Estudiante de Ingeniería en software
Responsabilidades	Encargado de la creación del documento de ERS cumpliendo las expectativas del sistema solicitado y también las normas IEE830
Información de contacto	Numero de contacto: 0992905126 Correo: iaescobar3@espe.edu.ec
Aprobación	

1.4 Definiciones, acrónimos y abreviaturas

Definiciones:

1. Normar ISO:

- Las Normas ISO son estándares internacionales desarrollados y publicados por la Organización Internacional de Normalización (ISO).
- La norma IEE830:1998, un documento crucial en el desarrollo de software que describe y precisa los requisitos que debe cumplir el software que se desarrollará.

2. Sistema: Se refiere a un software o aplicación diseñada para satisfacer ciertas necesidades o requisitos.

3. Framework: Es un software que ya tiene un diseño predeterminado dependiendo para que se necesite y que problema se resuelva

4. Buffer: Es una memoria lo cual permite almacenar datos de manera temporal que evita quedarse sin datos

5. Kernel: Es un sistema que es muy fundamental para el sistema operativo que permite ejecutarse en los modos de privilegios

6. Socket: El socket permite una comunicación que permite enviar o recibir los paquetes a través de la red

7. Cache: El cache es una memoria que esta entre el CPU y la RAM la cual permite el intercambio de datos

8. Endpoint: Son los puntos finales que se usan en la interfaz

9. Dashboard: Una interfaz gráfica que permite al usuario utilizar el sistema operativo.

Acrónimos:



Especificación de requisitos de software

1. **ERS:** Especificación de requisitos del sistema
2. **CPU:** Unidades de Procesamiento Central
3. **RAM:** Memoria de Acceso Aleatorio
4. **SO:** Sistema Operativo

Abreviaturas:

1. **Mb:** Megabyte
2. **Gb:** Gigabyte
3. **Temp:** Temperatura

1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
(IEEE,2008)	IEEE Std. 830-1998	https://www.fdi.ucm.es/profesor/gmendez/do_cs/is0809/ieee830.pdf	22/10/2008	IEEE
(Arsys, 2025)	¿Qué es un framework en programación y para qué sirve?	https://www.arsys.es/blog/que-es-un-framework-en-programacion-y-para-que-sirve#:~:text=Un%20framework%2C%20traducido%20habitualmente%20al,m%C3%A1s%20organizada%2C%20robusta%20y%20escalable.	02/03/2024	Arsys
(García, 2021)	Buffers y streams	https://es.linkedin.com/learning/fundamentos-esenciales-de-la-programacion-2/buffers-y-streams#:~:text=Un%20buffer%20es%20un%20espacio,punto%20de%20memoria%20a%20otro.	17/03/2021	García Gabriela
(Soto, 2020)	¿Qué es el Kernel y para qué sirve?	https://www.geeknetic.es/Kernel/que-es-y-para-que-sirve	26/07/2020	Soto Juan
(TylerMSFT, 2023)	Windows Sockets: Fondo	https://learn.microsoft.com/es-es/cpp/mfc/windows-sockets-background?view=msvc-170	16/06/2023	Tyler MSFT
(Amazon, 2024)	Información general sobre el almacenamiento en caché	https://aws.amazon.com/es/caching/	2024	Amazon
(Ibm, 2024)	¿Qué es un endpoint de API?	https://www.ibm.com/mx-es/topics/api-endpoint#:~:text=Un%20endpoint%20de%20API%20es,o%20localizadores%20unifor%20de%20recursos.	11/10/2024	IBM



Especificación de requisitos de software

(Ortiz & Cyberclick, 2023)	¿Qué es un dashboard y para qué se usa?	https://www.cyberclick.es/numerical-blog/que-es-un-dashboard	24/11/2023	Ortiz Dany
----------------------------	---	---	------------	------------

1.6 Resumen

El siguiente documento establecerá una descripción general de producto, en la que se incluirá la perspectiva del producto, en las que se especificara las funcionalidades del producto especificadas en requisitos funcionales, estas especificaran las capacidades y comportamiento que el sistema debe implementar para cumplir su objetivo que es el monitoreo de recursos para entorno de Linux, también en este presente documento se dedicara una sección a los requisitos no funcionales, donde estos son criterios de calidad que son fundamentales para garantizar que el sistema no solo cumpla con sus funcionalidades básicas, sino que también satisfaga los estándares de calidad de los esperados.

2 Descripción general

2.1 Perspectiva del producto

El sistema “*LINXPECTOR*” es un producto independiente donde no formara parte de ningún programa externo, este sistema se operará en entornos Linux ya sea dentro de un entorno virtual controlado o de un sistema Linux, el sistema interactuara con: sistema operativo Linux, sensores y métricas del sistema, Navegadores web de los usuarios y base de datos para el almacenamiento de históricos, usuarios y reportes.

2.2 Funcionalidad del producto

El sistema “*LINXPECTOR*” ofrecerá un monitoreo en tiempo real de: Uso de la CPU, utilización de la memoria RAM, tráfico de RED, Procesos en ejecución y almacenamiento en disco donde los datos recopilados se visualizarán en gráficos interactivos mostrando las estadísticas en tiempo real y mostrando el histórico de rendimiento, también el sistema tendrá la funcionalidad de unas notificaciones visuales para las alertas de umbrales configurados.

2.3 Características de los usuarios

Tabla 4 Usuario Administrador

Tipo de usuario	Administrador
Formación	Educación Secundaria
Habilidades	Conocimientos de gestión usuarios en plataformas web
Actividades	<ul style="list-style-type: none">• Acceso al sistema completo.• Análisis de datos mediante dashboard• Gestionar usuarios existentes en la plataforma

Tabla 5 Usuario



Especificación de requisitos de software

Tipo de usuario	Usuario registrado
Formación	Educación Secundaria
Habilidades	Habilidades básicas de computación en entorno linux .
Actividades	<ul style="list-style-type: none">• Ingreso al sistema con posibilidad de generar reportes.• Consulta de historial de reportes generados o guardados.• Configuración de umbrales para el sistema de alertas.

2.4 Restricciones

A continuación, se detalla las restricciones a considerar en el diseño y desarrollo del sistema de monitoreo "LINXPECTOR":

1. Restricciones Tecnológicas

- Lenguaje de programación:** Python como lenguaje principal
- Framework:** Django para la arquitectura del sistema
- Sistema Operativo:** Exclusivamente compatible con Linux (no funcionará en Windows o macOS)

2. Restricciones de Arquitectura

- Modelo Cliente-Servidor:** La aplicación debe seguir este patrón para permitir monitoreo remoto
- Tiempo Real:** La arquitectura debe soportar actualización de datos en tiempo real
- Base de Datos:** Se almacenará datos exclusivos del usuario en MySQL
- Escalabilidad:** Debe soportar monitoreo de múltiples servidores Linux simultáneamente

3. Restricciones de Desarrollo

- Control de versiones:** Git obligatorio para el control de código fuente
- Documentación:** Código documentado siguiendo PEP 8 y docstrings
- Testing:** Pruebas unitarias para componentes críticos del sistema

4. Restricciones de Idioma

- El sistema solo estará disponible en un idioma específico, como el español, lo que podría limitar su usabilidad para usuarios que no hablan ese idioma.

5. Restricciones de Seguridad

- Autenticación:** Sistema de login seguro con roles y permisos



- b. **Cifrado:** Comunicaciones cifradas entre componentes del sistema

2.5 Suposiciones y dependencias

Se asume que la disponibilidad de una conexión de red estable entre el servidor donde se aloja la plataforma y los servidores monitorizados, así como permisos de administrador en los sistemas objetivo para acceder a información privilegiada, cualquier desconexión de la red podría afectar al sistema ya que este utiliza librerías que son dependientes de que se tenga una buena conexión a internet.

2.6 Evolución previsible del sistema

A futuro, “LINXPECTOR” podría expandirse para incluir capacidades de monitoreo en otros sistemas operativos como Windows y macOS, implementar análisis predictivo mediante machine learning para anticipar fallos del sistema, incorporar un sistema de alertas proactivas basado en patrones históricos, y desarrollar una aplicación móvil dedicada para monitoreo en movimiento.

3 Requisitos específicos

3.1 Requisitos comunes de los interfaces

Esta sección detalla las entradas y salidas del sistema de software de “LINXPECTOR”, esto es fundamental para garantizar que el sistema interactúe de manera efectiva con los usuarios, el hardware y si existe, otros productos de software utilizados en la operación diaria.

3.1.1 Interfaces de usuario

En el sistema “LINXPECTOR” contara con una interfaz de usuario diseñada para ser intuitiva y fácil de usar. El diseño debe reflejar la identidad visual de una página de monitoreo en tiempo real, utilizando colores llamativos para la perspectiva del usuario, como lo es anaranjado y negro.

La disposición de la interfaz debe permitir a los usuarios no registrados utilizar correctamente el sistema sin la opción de poder ver reportes, analizar reportes o enviar alertas de umbrales configurados, así como el usuario autenticado podrá configurar umbrales de alertas de monitoreo para la CPU, RAM y RED emitiéndose una alerta al superar la configuración guardada por el usuario y el administrador tendrá la opción de gestionar a los usuarios dentro de la plataforma.

Para ello se ha creado plantillas donde se puede observar los mockups creados para nuestro sistema:

3.1.2 Interfaces de hardware

Las características que tiene nuestra máquina virtual son de 8gb de RAM es lo que habíamos designado que va a utilizar, de ahí utiliza el CPU 2



Especificación de requisitos de software

núcleos y 4 hilos lo cual le permite funcionar de mejor manera, por último, el almacenamiento que tiene es de 20GB que se lo designo no mucho porque todo se va a hacer con Python, con las diferentes herramientas para que funcione la interfaz correctamente

3.1.3 Interfaces de software

El sistema “LINXPECTOR” es un sistema autónomo que no requiere integración con otros productos de software externos, todas las funcionalidades necesarias para la operación del negocio están contenidas dentro del sistema, eliminando la necesidad de interfaces de software con sistemas de terceros.

3.1.4 Interfaces de comunicación

En este caso de la comunicación en la interfaz es la interfaz que hacemos principalmente la cual le permite comunicarse con el CPU, la red y el disco duro de cuanto se está utilizando mientras se va utilizando la máquina.

3.2 Requisitos funcionales

3.2.1 RF-01: Monitoreo de CPU

Número de requisito	RF-01
Nombre de requisito	Monitoreo de CPU
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe implementar un espacio de monitoreo continuo del procesador (CPU) que capture, analice y que presente métricas clave del CPU, este espacio o modulo es fundamental para capturar el rendimiento del sistema y prevenir situaciones de sobrecarga.

Entradas:

1. Datos del sistema sobre el uso de la CPU.
2. Datos de los sensores incorporados del sistema de la temperatura.
3. Información frecuente del trabajo.

Salidas:

1. Porcentaje en tiempo real del estado de CPU
2. Temperatura actual del CPU
3. Historial de calores

3.2.2 RF-02: Monitoreo de Memoria RAM

Número de requisito	RF02
Nombre de requisito	Monitoreo de Memoria RAM
Tipo	Requisito
Fuente del requisito	



Especificación de requisitos de software

Prioridad del requisito	Alta/Esencial
-------------------------	---------------

Descripción:

El sistema debe proporcionar una sección dedicado para el monitoreo completo del estado de la memoria RAM, ofreciendo una vista detallada de la utilización de memoria y su distribución en diferentes categorías.

Entradas:

1. Datos del sistema sobre la memoria total
2. Datos del uso actual de la memoria
3. Información de buffers, cache, factor de forma, velocidad de los módulos y ranuras utilizadas

Salidas:

1. Memoria total instalada
2. Memoria en uso
3. Memoria libre
4. Uso de cache, factor de forma, velocidad de módulos y ranuras utilizadas
5. Gráficos dinámicos en tiempo real

3.2.3 RF-03: Monitoreo de Red

Número de requisito	RF-03
Nombre de requisito	Monitoreo de Red
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe monitorear el uso del tráfico de la red, incluyendo métricas de rendimiento y detección de problemas en las interfaces de red.

Entradas:

1. Datos de tráfico de red entrante y saliente.
2. Estados de interfaces de red.

Salidas:

1. Velocidad de transmisión y recepción
2. Uso total de red
3. Gráficos dinámicos en tiempo real del uso de la red.

3.2.4 RF-04: Monitoreo de Procesos

Número de requisito	RF-04
Nombre de requisito	Monitoreo de procesos en ejecución
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:



Especificación de requisitos de software

El sistema debe proporcionar la capacidad de monitoreo detallado de todos los procesos de ejecución, mostrando y analizando su impacto de los recursos del sistema como la Memoria, CPU y RAM

3.2.5 RF-05: Monitoreo de Disco Duro

Número de requisito	RF-05
Nombre de requisito	Monitoreo de Disco Duro
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe permitir la visualización de las funcionalidades para el monitoreo completo del almacenamiento en disco, proporcionando información detallada sobre el uso y disponibilidad de espacio.

Entrada:

1. Capacidad total del disco.
2. Datos del uso actual
3. Información de particiones.

Salidas:

1. Espacio total
2. Espacio usado
3. Espacio libre
4. Gráfico de utilización.

3.2.6 RF-06: Dashboard

Número de requisito	RF-06
Nombre de requisito	Dashboard
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe incluir la visualización de gráficos del monitoreo del sistema, donde permita realizar al usuario un informe sobre el uso los datos recopilados del CPU, RAM y RED.

Entrada:

1. Información sobre la RAM
2. Información sobre la RED
3. Información sobre el CPU
4. Información sobre el Disco Duro.

Salidas:

1. Gráfico del uso de la RAM
2. Gráfico de la entradas y salidas de la RED
3. Gráfico del uso de la CPU
4. Gráfico del uso del Disco
5. Informe sobre el monitoreo del sistema.



3.2.7 RF-07: Gráficos en tiempo real

Número de requisito	RF-07
Nombre de requisito	Gráficos en tiempo real
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe incluir gráficos dinámicos donde muestren las métricas monitoreadas en tiempo real.

Entrada:

1. Datos del monitoreo en tiempo real
2. Preferencias de visualización

Salidas:

1. Gráficos actualizados en tiempo real
2. Opciones personalizadas

3.2.8 RF-08: Sistema de Alertas

Número de requisito	RF-08
Nombre de requisito	Sistema de Alertas
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe incluir un mecanismo de alertas que notifique al usuario sobre las condiciones críticas de la RAM, RED o CPU.

Entradas:

1. Umbrales predefinidos por el usuario
2. Datos del monitoreo en tiempo real
3. Configuración de alertas.

Salidas:

1. Alertas visuales
2. Notificación en tiempo real
3. Registro de alertas

3.2.9 RF-09: Autenticación de Usuarios

Número de requisito	RF-09
Nombre de requisito	Autenticación de Usuarios
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:



El sistema debe permitir una autenticación segura de los usuarios, para así poder acceder a las funcionalidades adicionales de la plataforma de monitoreo.

Entradas:

1. Credenciales del usuario
2. Datos de perfil
3. Permisos o roles asignados

Salidas:

1. Acceso autorizado o denegado
2. Sesión de usuario
3. Registro o historia de actividades

3.2.10 RF-10: Configuración de Umbrales

Número de requisito	RF-10
Nombre de requisito	Configuración de Umbrales
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe permitir a los usuarios configurar umbrales personalizados para las alertas en los diferentes recursos monitoreados.

Entrada:

1. Preferencia de usuario
2. Valores de umbrales personalizados
3. Configuración de alertas.

Salidas:

1. Confirmación de cambios
2. Vista previa de las configuraciones

3.2.11 RF-11: Almacenamiento de Datos

Número de requisito	RF-11
Nombre de requisito	Almacenamiento de Datos
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe permitir guardar los datos personalizados de cada usuario en la base de datos según las alertas

Entrada:

1. Preferencia de usuario
2. Valores de alerta personalizados
3. Configuración de alertas.
4. Enviar a la base de datos cada cierto designado por el usuario

Salidas:

5. Confirmación de envió
6. Vista de datos en la base de datos



3.2.12 RF-12: Interfaz de Usuario

Número de requisito	RF-12
Nombre de requisito	Interfaz de Usuario
Tipo	Requisito
Fuente del requisito	
Prioridad del requisito	Alta/Esencial

Descripción:

El sistema debe tener una interfaz que le permita al usuario poder entender bien la interfaz de cómo funciona

Entrada:

1. Login del usuario
2. La interfaz

Salidas:

3. Ingreso a la interfaz
4. Vista de la interfaz con todo lo que contiene.

3.3 Requisitos no funcionales

3.3.1 RNF-01: Requisitos de rendimiento

Para que pueda soportar la carga el sistema en este caso se está usando un entorno virtual instalando Linux, los recursos que se utilizan son 8gb de RAM, 2 núcleos con 4 hilos en la cpu y de almacenamiento 20gb mientras va controlando el tráfico de red.

Las herramientas que vamos a utilizar en nuestra máquina virtual son: Linux, chart.js, Django y Python para poder usar todas las herramientas.

3.3.2 RNF-02: Seguridad

Se debe implementar medidas de seguridad para la protección de los datos e información de los usuarios registrados, así como su Gmail, contraseña que es lo que se utiliza para la autenticación.

El sistema no debe permitir que los datos de los usuarios sean públicos.

3.3.3 RNF-03: Fiabilidad

Lo que se necesita en el sistema es la máquina virtual porque estamos haciendo en un entorno controlado con Linux

3.3.4 RNF-04: Disponibilidad

Hay que asegurar que la plataforma esté disponible en todo momento, con un mínimo tiempo de inactividad planificado. Implementar medidas para recuperarse rápidamente de posibles fallas o interrupciones.



3.3.5 RNF-05: Mantenibilidad

El sistema debe ser desarrollado siguiendo principios de código limpio y buenas prácticas de programación que faciliten su mantenimiento a largo plazo. La arquitectura debe ser modular, con componentes bien definidos y desacoplados que permitan modificaciones o ampliaciones sin afectar otras partes del sistema. Además, se implementará un sistema de control de versiones (Git).

3.3.6 RNF-06: Portabilidad

El sistema debe ser diseñado considerando la portabilidad entre diferentes distribuciones de Linux, garantizando su funcionamiento en las versiones más comunes como Ubuntu, CentOS, Debian y Red Hat Enterprise Linux.

3.4 Otros requisitos

No es necesario

4 Apéndices

A continuación, se presentará links de documentos creados para una mayor comprensión

4.1 GitHub del Sistema

<https://github.com/elkinpabon/Plataforma-web-para-analisis-de-recursos-Linux>

4.2 Mockups del Sistema

<https://github.com/elkinpabon/Plataforma-web-para-analisis-de-recursos-Linux/blob/main/1.%20Documentation/Mockups%20del%20Sistema.pdf>

4.3 Diagrama Entidad Relación

[Plataforma-web-para-analisis-de-recursos-Linux/2. UML Diagrams/Diagrama Entidad Relación.pdf at main · elkinpabon/Plataforma-web-para-analisis-de-recursos-Linux · GitHub](#)

4.4 Diagrama de caso de uso

[Plataforma-web-para-analisis-de-recursos-Linux/2. UML Diagrams/Diagrama de Caso de Uso.pdf at main · elkinpabon/Plataforma-web-para-analisis-de-recursos-Linux · GitHub](#)