



Universidad de las Ciencias  
Informáticas

**Universidad de las Ciencias Informáticas**

**Facultad 4**

# **Sistema de visualización de indicadores para la vigilancia tecnológica**

**Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas**

Autor:

Sandy Valiente Rodríguez

Tutor:

Dr. C. Omar Correa Madrigal

**La Habana, noviembre 2023  
“Año 65 de la Revolución”**

## **Dedicatoria**

*A mis padres y hermana, por su amor incondicional, apoyo inquebrantable y sacrificio incansable. A todos mis profesores, por su guía, sabiduría y paciencia. A mis amigos, por su ánimo, comprensión, alegría y buenos momentos compartidos. A todos aquellos que me han inspirado y alentado en este camino. Esta tesis está dedicada a ustedes, con todo mi cariño y agradecimiento.*

## **Agradecimientos**

*Quiero expresar mi profundo agradecimiento a todas las personas que contribuyeron a la realización de esta tesis. En primer lugar, agradezco a mi tutor por su orientación, paciencia y apoyo a lo largo de este proceso. También quiero agradecer a mis profesores y asesores por su valiosa retroalimentación y consejos. Agradezco a mis compañeros de clase por su colaboración y motivación. A mi familia y amigos, les agradezco por su comprensión, apoyo emocional y palabras de aliento. A mi casa de altos estudios, que me ha permitido superarme y llegar hasta este punto. Finalmente, agradezco a todas las personas que participaron en mi investigación y a quienes generosamente compartieron su tiempo y conocimientos. Su contribución ha sido fundamental para el desarrollo de este trabajo. Muchas gracias a todos.*

## Declaración de autoría

Queda declarada la autoría de la presente tesis titulada: “Sistema de visualización de indicadores para la vigilancia tecnológica”, y se reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmo la presente a los 24 días del mes de noviembre del año 2023.



Autor: Sandy Valiente Rodríguez



Tutor: Dr. C. Omar Correa Madrigal

## Resumen

En esta investigación se parte de la existencia de un estudio, en la Universidad de las Ciencias Informáticas, para poner en marcha el desarrollo de un ecosistema para la vigilancia tecnológica, y su implementación en un observatorio tecnológico, que proporcione a sus usuarios información actualizada, mecanismos de alertas y facilidad en la búsqueda de información con un comportamiento proactivo. A pesar de todas estas facilidades que brinda, dicho observatorio tecnológico carece de funcionalidades de monitoreo sobre tendencias e indicadores relacionados con temas de investigación. Se propone utilizar elementos de inteligencia de negocios, como son los *dashboards*, para modelar dicho observatorio con nuevas funcionalidades que posibiliten el análisis y visualización de datos. El objetivo de la investigación consiste en desarrollar un sistema que permita el monitoreo de la información de temas de investigación a partir del uso de indicadores que posibiliten el análisis y visualización de la información. Para ello, se usa una herramienta automatizada de visualización. El sistema de visualización permite el análisis sobre tecnologías de desarrollo de software, como lenguajes de programación, mediante el uso de indicadores que exponen la popularidad de la tecnología a lo largo del tiempo. De esta manera, se da un primer paso hacia la escalabilidad del sistema. Se emplean los métodos de investigación analítico-sintético, inductivo, análisis documental y modelado. Para el desarrollo de la investigación se emplea la metodología ágil Extreme Programming (XP).

**Palabras clave:** análisis de datos, *dashboard*, observatorio tecnológico, vigilancia tecnológica, visualización de datos.

## **Abstract**

*This research is based on the existence of a study at the University of Informatic Sciences, to launch the development of an ecosystem for technological surveillance, and its implementation in a technological observatory, which provides its users with updated information, alert mechanisms and ease in the search for information with a proactive behavior. Despite all these facilities, such a technology observatory lacks monitoring functionalities on trends and indicators related to research topics. It is proposed to use business intelligence elements, such as dashboards, to model this observatory with new functionalities that enable data analysis and visualization. The objective of the research is to develop a system that allows the monitoring of information on research topics through the use of indicators that enable the analysis and visualization of the information. For this purpose, an automated visualization tool is used. The visualization system allows the analysis of software development technologies, such as programming languages, through the use of indicators that expose the popularity of the technology over time. In this way, a first step towards scalability of the system is taken. Analytical-synthetic, inductive, documentary analysis and modelling research methods are employed. The agile methodology Extreme Programming (XP) is used for the development of the research.*

**Keywords:** *dashboard, data analysis, data visualization, technology observatory, technology watch.*

# Índice

Introducción.....	1
Capítulo I. Fundamentación teórica sobre análisis y visualización de indicadores para vigilancia tecnológica.....	6
Introducción al capítulo.....	6
1.1. Conceptos asociados a la investigación.....	6
1.1.1. Vigilancia tecnológica.....	6
1.1.2. Observatorio tecnológico.....	6
1.1.3. Análisis de datos.....	7
1.1.4. Visualización de datos.....	9
1.1.5. Dashboard.....	12
1.1.6. Indicador clave de rendimiento KPI.....	13
1.2. Sistemas de visualización sobre tecnologías de desarrollo de software.....	14
1.2.1. TIOBE Index.....	15
1.2.2. PYPL PopularitY of Programming Language Index.....	18
1.3. Metodología de desarrollo de software.....	24
1.4. Herramientas y tecnologías empleadas en el desarrollo de la propuesta de solución.....	27
1.4.1. Herramienta de visualización.....	27
1.4.2. Herramienta CASE.....	31
1.4.3. Lenguaje de modelado.....	31
1.4.4. Entorno de desarrollo integrado.....	32
1.4.5. Lenguaje de programación.....	33
1.4.6. Gestor de base de datos.....	35
1.4.7. Herramientas de pruebas de software.....	37
1.4.8. Fuentes de información.....	40
Conclusiones parciales.....	40
Capítulo II. Análisis y diseño de un sistema de visualización de indicadores para la vigilancia tecnológica.....	42
Introducción al capítulo.....	42

2.1. Descripción de la propuesta de solución.....	42
2.1.1. Arquitectura cliente-servidor.....	42
2.1.2. Patrón arquitectónico .....	44
2.1.3. Principales módulos.....	47
2.1.4. Flujo de información.....	48
2.1.5. Arquitectura de navegación.....	49
2.2. Requisitos del sistema .....	50
2.2.1. Requisitos informacionales .....	50
2.2.2. Requisitos funcionales .....	52
2.2.3. Requisitos no funcionales .....	55
2.3. Historias de usuario .....	57
Conclusiones parciales .....	61
Capítulo III. Implementación y prueba del sistema de visualización de indicadores para la vigilancia tecnológica .....	62
Introducción al capítulo .....	62
3.1. Funcionamiento del sistema de visualización.....	62
3.1.1. Descripción de los contenidos de la página principal .....	62
3.1.2. Descripción de los contenidos de la página Paneles de control .....	63
3.1.3. Descripción de los contenidos de la página Usuarios .....	64
3.1.4. Descripción de los contenidos de la página Dashboard .....	65
3.1.5. Diagrama de despliegue .....	66
3.2. Estrategia de pruebas .....	67
3.3. Diseño de casos de prueba por tipo y resultados.....	68
3.3.1. Pruebas alfa .....	68
3.3.2. Pruebas de rendimiento.....	75
3.3.3. Pruebas de seguridad.....	77
3.3.4. Pruebas de portabilidad .....	79
Conclusiones parciales .....	81
Conclusiones generales .....	82



Recomendaciones..... 83

Referencias bibliográficas ..... 84

## Introducción

Se suele decir que “una imagen vale más que mil palabras”. Hoy, en la era del *big data*, cuando las empresas se inundan con información de diversos tipos de datos que provienen de fuentes basadas en la nube, examinar la información para comprender lo que importa y lo que no es cada vez más difícil. Las imágenes hacen que el análisis sea mucho más fácil y rápido y ofrecen la posibilidad de ver lo importante. Es más, la mayoría de las personas responden mucho mejor a las imágenes que al texto: el 90 % de la información que recibe el cerebro es visual, y este procesa las imágenes 60 000 veces más rápido que el texto. Esos puntos son un argumento convincente para el uso de la visualización de datos en el análisis y la transmisión de información (1).

A medida que la era del *big data* entra en pleno apogeo, la visualización es una herramienta cada vez más importante para darle sentido a los datos que se generan cada día. La visualización de datos ayuda a contar historias seleccionando los datos en una forma más fácil de entender, destacando las tendencias y los valores atípicos. Una buena visualización cuenta una historia, eliminando el ruido de los datos y resaltando la información útil (2).

La visualización de datos no es un concepto nuevo. La primera forma de visualización de datos se remonta a los egipcios antes del siglo XVII, en gran parte utilizada para la asistencia en la navegación. Las pinturas en las paredes de la cueva de Lascaux podrían considerarse una forma de visualización de datos que narra historias de caza de hace muchos miles de años. Con el paso del tiempo, se ampliaron las aplicaciones de las visualizaciones de datos, como en disciplinas económicas, sociales y sanitarias. La alta tecnología ha introducido nuevas opciones visuales (1, 3).

Según la norma UNE 166006:2018 Gestión de la I+D+i, la vigilancia tecnológica es un proceso organizado, selectivo y permanente, de captar información del exterior y de la propia organización sobre ciencia y tecnología, seleccionarla, analizarla, difundirla y comunicarla, para convertirla en conocimiento para tomar decisiones con menor riesgo y poder anticiparse a los cambios (4). La vigilancia tecnológica aporta a la inteligencia competitiva, la cual es vital para cualquier organización. Se entiende como inteligencia competitiva a la recolección y análisis sistemático de información desde múltiples fuentes y a través de un programa. Al soportar los procesos estratégicos de una organización, como un sensor que muestra a los directivos si la organización avanza o no, si es competitiva o no, la vigilancia tecnológica deviene en un proceso estratégico y herramienta fundamental para el desarrollo (5).

En Cuba, los sistemas de vigilancia tecnológica se destacan en el sector de la educación

superior, y en el empresarial, donde se demuestra la importancia y la repercusión que tienen en el marco de los sistemas de gestión de I+D+i. El apoyo gubernamental para una mayor introducción de los resultados de la ciencia, y con ello lograr la innovación tecnológica, organizacional y social, han llevado a que se promueva la práctica de la vigilancia tecnológica, sustentada en observatorios. En el país existen instituciones que apoyan este proceso de mayor impacto de la ciencia y la tecnología como: el Instituto de Información Científico Tecnológica del CITMA, la Empresa INFOMED del MINSAP, ETI de Biocubafarma, y la Red de Observatorios del MES. En este contexto, se han desarrollado importantes eventos científicos como: Congreso Internacional de Información y el Taller Internacional InfoPolos en la Convención Internacional de Ciencia, Tecnología e Innovación. Ambos han permitido posicionar y discutir importantes aspectos del desarrollo de la vigilancia tecnológica en el país y el mundo (5).

La industria de software en Cuba adolece de observatorios que apoyen los procesos de vigilancia tecnológica y las mejores experiencias se circunscriben al análisis de inteligencia en relación al mercado. Los aspectos científico-técnicos de la práctica ingenieril del desarrollo de software, han avanzado desde el trabajo de grupos de expertos de universidades, empresas y centros de investigación. Ante un escenario tecnológico tan cambiante y de alta competitividad como el de las Tecnologías de la Información y las Telecomunicaciones (TICs), resulta necesario complementar la toma de decisiones con productos de vigilancia científico tecnológica como cuadros de mandos digitales cuyos resultados se expresen en la integración y uso intensivo de tecnologías computacionales (5).

La Universidad de las Ciencias Informáticas en conjunto con el Ministerio de las Comunicaciones, con el fin de favorecer el desarrollo de la soberanía tecnológica en el país, han decidido poner en marcha el desarrollo de un ecosistema para la vigilancia tecnológica, y su implementación en un observatorio tecnológico. Los sistemas que componen este ecosistema tecnológico realizan búsquedas especializadas a través de internet usando como base múltiples fuentes de información como algunos motores de búsqueda, repositorios, bibliotecas y bases de datos académicas. En este sentido, se presentaron las bases de una herramienta que permita a los especialistas realizar búsquedas sobre un tema determinado, así como la gestión de los resultados para análisis de la vigilancia tecnológica. A pesar de todas estas funcionalidades que brinda, dicho observatorio tecnológico carece de un sistema de monitoreo sobre tendencias e indicadores relacionados con temas de investigación. En este sentido, resulta necesario desarrollar un sistema para dotar al observatorio tecnológico de la capacidad de monitoreo a través de la visualización de datos.

## **Problema de investigación**

¿Cómo dotar al ecosistema de vigilancia tecnológica de funcionalidades para la visualización de indicadores de investigación?

## **Objeto de estudio**

Está centrado en los sistemas de visualización de datos.

## **Objetivo general**

Desarrollar un sistema web para el monitoreo de la información de temas de investigación a partir de indicadores que permitan el análisis y visualización de datos.

## **Campo de acción**

Métodos, técnicas y herramientas computacionales para el monitoreo, análisis y visualización de datos.

## **Tareas de investigación**

- Análisis del estado del arte de los distintos métodos, técnicas y herramientas de software especializadas en el análisis y visualización de datos.
- Estudio y caracterización de sistemas homólogos, identificando los principales elementos que los componen.
- Caracterización de las herramientas y tecnologías que se utilizarán en el desarrollo de solución.
- Análisis y diseño de un sistema de visualización de indicadores para vigilancia tecnológica.
- Implementación de un sistema de visualización de indicadores para la vigilancia tecnológica de temas de investigación, utilizando gráficos como medios de visualización.
- Ejecución de pruebas de software sobre el sistema de visualización.

## **Métodos de investigación**

La metodología de esta investigación se basa, en gran medida, en la recopilación, síntesis y análisis de diversas fuentes académicas.

### **Métodos teóricos:**

- Analítico-sintético: uno de los métodos que sigue este trabajo es analítico-sintético, a través de la revisión de bibliografía sobre la temática abordada. Se emplea para realizar un estudio de los documentos y herramientas más relevantes sobre el uso e importancia de la visualización de datos, permitiendo así, extraer los elementos más importantes que sirvan de sustento para la investigación.
- Análisis documental: se analizan referentes teóricos acerca del tema de investigación para seleccionar los aspectos teóricos - metodológicos para la obtención de los resultados de la investigación.
- Inductivo: el método inductivo está presente en los resultados y conclusiones obtenidas, partiendo de hechos aceptados como válidos.
- Modelación: se emplea para la confección del sistema de visualización de indicadores para vigilancia tecnológica, al tener en cuenta los elementos teóricos y prácticos que lo caracterizan a partir del sustento teórico que aporta la investigación.

### **Estructura del documento**

La investigación se compone de: introducción, desarrollo compuesto por tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas. La introducción inicia con una presentación del problema o desafío técnico que se aborda. Esto incluye una descripción clara y concisa del problema específico que se pretende resolver o investigar, junto con su importancia en el contexto en que se desenvuelve. Se presenta el objetivo que la investigación pretende lograr, junto con una delimitación clara de su alcance. Se describe el enfoque metodológico que se utiliza para abordar el problema, incluyendo los métodos de investigación que se emplean. El primer capítulo, como inicio del desarrollo, hace referencia a la fundamentación teórica de la investigación. Es la base conceptual y teórica sobre la cual se sustenta el estudio. Consiste en una revisión exhaustiva y crítica de las teorías, conceptos e investigaciones previas relacionadas con el tema de investigación. Define las herramientas a utilizar para el desarrollo de una propuesta. El segundo capítulo trata sobre el proceso de análisis y diseño del sistema de visualización. Consiste en identificar, comprender y definir las necesidades y requisitos del sistema, y luego transformar esos requisitos en una solución del

software bien estructurada y diseñada. Por último, el tercer capítulo describe el proceso de implementación del sistema y pruebas de software sobre el mismo. Consiste en llevar a cabo la transformación de la especificación del software en un sistema funcional y realizar pruebas exhaustivas para verificar su correcto funcionamiento. Las conclusiones generales constituyen una sección crucial que resume y presenta de manera concisa los hallazgos, las implicaciones y las contribuciones más significativas derivadas del estudio realizado. Las recomendaciones constituyen una sección importante donde se brindan sugerencias lógicas y prácticas basadas en los hallazgos del estudio. Estas recomendaciones están diseñadas para ofrecer orientación aplicable y acciones específicas que pueden derivarse de la investigación realizada. Las referencias bibliográficas es una sección fundamental que enumera todas las fuentes consultadas, citadas o referenciadas a lo largo del documento. Esta sección juega un papel crucial en la validación y respaldo de los argumentos presentados, además de cumplir con estándares académicos y éticos.

# **Capítulo I. Fundamentación teórica sobre análisis y visualización de indicadores para vigilancia tecnológica**

## **Introducción al capítulo**

El estudio del estado del arte es una investigación exhaustiva y sistemática de las publicaciones, investigaciones y avances más recientes y relevantes en un campo específico de conocimiento o actividad. En el presente capítulo se exponen los referentes fundamentales y conceptos del trabajo investigativo. Estos buscan el fin de facilitar la comprensión del objeto de estudio, así como los análisis derivados de la investigación para una propuesta coherente del modelo. Se describen, además, las principales herramientas y tecnologías que se utilizaron para llevar a cabo el análisis, diseño e implementación de la propuesta de solución.

## **1.1. Conceptos asociados a la investigación**

### **1.1.1. Vigilancia tecnológica**

Según la UNE 166006, la vigilancia tecnológica es un proceso organizado, selectivo y permanente de captar información del exterior y de la propia organización sobre ciencia y tecnología, para convertirla en conocimiento para tomar decisiones con menor riesgo y poder anticiparse a los cambios (4). Para la Norma Cubana NC 1308 es un proceso ético y sistemático de recolección y análisis de información acerca del ambiente de negocios, de los competidores y de la propia organización, y comunicación de su significado e implicaciones destinada a la toma de decisiones (6).

### **1.1.2. Observatorio tecnológico**

Un observatorio tecnológico se relaciona como un conjunto de acciones organizadas con el propósito de interpretar realidades, objetos, actores, reconstruir situaciones, variables, indicadores con el fin de procesar y generar información y datos, que abonen y apoyen la toma de decisiones. Los observatorios son un sistema estructurado y organizado en permanente búsqueda de información de calidad, validando su origen, haciendo análisis del entorno, monitoreándolo, construyendo uno nuevo, motivando su uso en el diseño de estrategias por otro. Es importante definir que un observatorio es una organización creada por un colectivo con el fin

de seguir la evolución de un fenómeno, generalmente de naturaleza económica o social, desde una posición favorable (7).

Para Sarmiento un observatorio es una herramienta para realizar vigilancia tecnológica, que reconoce cambios en el dominio de la información que procesa, gestiona y observa. Por lo tanto, teniendo en cuenta comportamientos previos, puede avisar con antelación de ciertas variaciones o diferencias en parámetros que evalúa, generando un conocimiento con un alto nivel de importancia al ser actual y novedoso, lo que puede ser utilizado por los receptores que tengan interés en esa información. Es un sitio web conformado por distintas herramientas que permiten crear un entorno informático, para el acceso versátil a una red distribuida de archivos de información y datos a través de Internet, y se constituyen como un sitio de reunión, que suma opiniones y colaboraciones relacionadas con un tema específico. Un observatorio tecnológico captura informaciones externas con el propósito de transformarlas en conocimientos específicos que conducen a sus usuarios a tomar decisiones (8).

### **1.1.3. Análisis de datos**

El análisis de datos convierte datos sin procesar en información práctica. Incluye una serie de herramientas, tecnologías y procesos para encontrar tendencias y resolver problemas mediante datos. Los análisis de datos pueden dar forma a procesos empresariales, mejorar la toma de decisiones e impulsar el crecimiento empresarial (9).

El análisis de datos sigue cinco pasos para analizar cualquier conjunto de datos de gran tamaño (9):

- Recopilación de datos.
- Almacenamiento de datos.
- Procesamiento de datos.
- Limpieza de datos.
- Análisis de datos.

### **Recopilación de datos**

Esto incluye identificar el origen de datos y recopilar datos de allí. La recopilación de datos sigue los procesos ETL o ELT.



- ETL: extracción, transformación y carga: en un proceso ETL, los datos generados se transforman primero en un formato estándar y luego se cargan en el almacenamiento (9).
- ELT: transformación de la carga extraída: en la ELT, los datos se cargan primero en la memoria y luego se transforman en el formato requerido (9).

## **Almacenamiento de datos**

Según la complejidad de los datos, estos se pueden trasladar a un almacenamiento como los almacenamientos de datos en la nube o los lagos de datos.

Un almacenamiento de datos es una base de datos optimizada para analizar datos relacionales procedentes de sistemas transaccionales y aplicaciones empresariales. La estructura y el esquema de los datos se definen previamente para optimizar la rapidez de la búsqueda y elaboración de informes. Los datos se limpian, enriquecen y transforman para que actúen como fuente en la que los usuarios puedan confiar (9).

Un lago de datos es diferente porque puede almacenar tanto datos estructurados como no estructurados sin ningún procesamiento posterior. La estructura de los datos o el esquema no se definen cuando se capturan los datos; esto significa que se pueden almacenar todos los datos sin necesidad de un diseño cuidadoso, lo que resulta muy útil cuando se desconoce el uso futuro de estos (9).

## **Procesamiento de datos**

Cuando los datos están en su sitio, hay que convertirlos y organizarlos para obtener resultados precisos de las consultas analíticas. Para ello, existen diferentes opciones de procesamiento de datos. La elección de la estrategia depende de los recursos informáticos y analíticos disponibles para su procesamiento (9).

- Procesamiento centralizado: todo el procesamiento se lleva a cabo en un servidor central dedicado que aloja todos los datos (9).
- Procesamiento distribuido: los datos se distribuyen y almacenan en diferentes servidores (9).
- Procesamiento por lotes: los datos se acumulan con el tiempo y se procesan por lotes (9).
- Procesamiento en tiempo real: los datos se procesan continuamente, y las tareas de

cálculo terminan en segundos (9).

## **Limpieza de datos**

La limpieza de datos implica la búsqueda de errores, como duplicaciones, incoherencias, redundancias o formatos incorrectos. Asimismo, se usa para filtrar los datos no deseados para los análisis (9).

## **Análisis de datos**

Este es el paso en el que los datos brutos se convierten en información procesable. A continuación, se describen cuatro tipos de análisis de datos.

- **Análisis descriptivo:** los científicos de datos se encargan de analizar estos datos para comprender lo que sucedió o sucede con ellos. Se caracteriza por la visualización de datos, como los gráficos circulares, de barras, lineales, tablas o narraciones generadas (9).
- **Análisis de diagnóstico:** el análisis de diagnóstico es un proceso de profundización o análisis detallado de los datos para entender por qué ocurrió algo. Se caracteriza por técnicas como el análisis detallado, el descubrimiento y la minería de datos o las correlaciones (9).
- **Análisis predictivo:** el análisis predictivo usa datos históricos para hacer previsiones precisas sobre las tendencias futuras. Se caracteriza por técnicas como el *machine learning*, la previsión, la coincidencia de patrones y el modelado predictivo (9).
- **Análisis prescriptivo:** el análisis prescriptivo lleva los datos predictivos al siguiente nivel. No solo predice lo que es probable que ocurra, sino que sugiere una respuesta óptima para ese resultado. Puede analizar las posibles implicaciones de las diferentes alternativas y recomendar el mejor curso de acción. Se caracteriza por el análisis de gráficos, la simulación, el procesamiento de eventos complejos, las redes neuronales y los motores de recomendación (9).

### **1.1.4. Visualización de datos**

La visualización de datos es la representación de datos mediante el uso de gráficos comunes, como gráficas, diagramas, infografías e incluso animaciones. Estas representaciones visuales de

información comunican relaciones de datos complejas y conocimientos basados en datos de una manera que resulta fácil de comprender (3). Para Amazon Web Services la visualización de datos es el proceso de utilizar elementos visuales como gráficos o mapas para representar datos. De esta manera, se trasladan datos complejos, de alto volumen o numéricos a una representación visual más fácil de procesar. Las herramientas de visualización de datos mejoran y automatizan el proceso de comunicación visual para lograr precisión y detalle (10).

Harvard Business Review categoriza la visualización de datos en cuatro propósitos principales: generación de ideas, ilustración de ideas, descubrimiento visual y visualización de datos diarios. A continuación, se describirán detalladamente cada uno de ellos.

- Generación de ideas: la visualización de datos se suele utilizar para estimular la generación de ideas en los equipos. Con frecuencia se aplican en sesiones de *brainstorming* (lluvia de ideas) o *design thinking* al inicio de un proyecto para apoyar la recopilación de diferentes perspectivas y resaltar los intereses comunes del grupo (3).
- Ilustración de ideas: la visualización de datos para la ilustración de ideas ayuda a transmitir una idea, como una táctica o un proceso. Se usa comúnmente en entornos de aprendizaje, como tutoriales, cursos de certificación o centros de excelencia, pero también se puede usar para representar estructuras o procesos de organización, facilitando la comunicación entre las personas más adecuadas para tareas específicas (3).
- Descubrimiento visual: el descubrimiento visual y la visualización de datos diarios están más estrechamente alineados con los equipos de datos. Si bien el descubrimiento visual ayuda a los analistas de datos, a los científicos de datos y a otros profesionales de datos a identificar patrones y tendencias dentro de un conjunto de datos, la visualización de datos diarios apoya la narración posterior después de que se haya encontrado nueva información (3).
- Visualización de datos: la visualización de datos es un paso crítico en el proceso de ciencia de datos, ya que ayuda a equipos y personas a nivel individual a transmitir datos de manera más efectiva a colegas y responsables de la toma de decisiones (3).

### **Tipos de visualizaciones de datos**

Los paneles de control son herramientas efectivas de visualización de datos para rastrear y visualizar datos de varios orígenes de datos, ya que brindan visibilidad sobre los efectos de comportamientos específicos por parte de un equipo o alguien adyacente sobre el rendimiento

(3). Los paneles de control incluyen técnicas comunes de visualización, tales como:

- Gráficos de columnas: este tipo de gráfico es útil para mostrar los cambios de los datos en un período de tiempo o para ilustrar comparaciones entre elementos. A menudo en estos gráficos las categorías se suelen organizar a lo largo del eje horizontal, mientras que los valores se representan a lo largo del eje vertical (11).
- Gráficos de barras: permite mostrar comparaciones entre elementos individuales. A menudo en estos gráficos las categorías se suelen organizar a lo largo del eje vertical, mientras que los valores lo hacen a lo largo del horizontal (11).
- Gráficos de líneas: permiten mostrar datos continuos en el tiempo, establecidos frente a una escala común y, por tanto, son ideales para mostrar tendencias en datos a intervalos iguales. En un gráfico de líneas, los datos de categoría se distribuyen uniformemente en el eje horizontal y todos los datos de valor se distribuyen uniformemente en el eje vertical (11).
- Gráficos circulares: muestran el tamaño de los elementos de una serie de datos, en proporción a la suma de los elementos. Los puntos de datos de un gráfico circular se muestran como porcentajes del total del gráfico circular (11).
- Tablas: las tablas organizan los datos en filas y columnas, se utilizan para mostrar conjuntos de datos bidimensionales que pueden ser organizados categóricamente (11).

### **Componentes de la visualización de datos**

Los científicos de datos combinan tres componentes principales para visualizar los datos.

- Historia: la historia representa el propósito de las visualizaciones de datos. El científico de datos se comunica con varias partes interesadas de acuerdo a lo que quieren conseguir mediante el análisis de datos. Los científicos de datos y los usuarios empresariales colaboran para identificar el tipo de historia que quieren que los datos les cuenten (10).
- Datos: a continuación, los analistas de datos identifican los conjuntos de datos adecuados que les permitirán narrar la historia de los datos. Modifican los formatos de datos existentes, limpian los datos, eliminan los valores atípicos y llevan a cabo otros análisis. Tras la preparación de los datos, planifican los diferentes métodos de exploración visual (10).
- Visuales: a continuación, los científicos de datos seleccionan los métodos de visualización

más adecuados para compartir los nuevos conocimientos. Crean cuadros y gráficos que destacan los puntos de datos clave y simplifican los conjuntos de datos complejos. Piensan en formas eficientes de presentar sistemáticamente los datos para la inteligencia empresarial (10).

### **1.1.5. Dashboard**

Stephen Few define un *dashboard* como una visualización de la información más importante necesaria para lograr uno o más objetivos; consolidado y organizado en una sola pantalla para que la información se pueda monitorear de un vistazo (12). Para Microsoft es una herramienta que numerosas empresas usan para seguir, analizar y mostrar datos, por lo general para obtener información del bienestar general de una organización, un departamento o un proceso específico (13).

Los *dashboards* de inteligencia de negocios son soluciones de administración de información y visualización de datos que se utilizan para analizar datos. Los autores de contenido pueden usar elementos interactivos como filtros y acciones para combinar gráficos e informes en una sola pantalla. De esta manera, los usuarios pueden conocer el panorama general al instante. Los *dashboards* son una de las funcionalidades más populares de las plataformas de BI. Esto se debe a que ofrecen análisis de datos fáciles de comprender, permiten personalizar qué información ver y proporcionan maneras de compartir los resultados del análisis con otros usuarios (14).

Los *dashboards* son un componente esencial de la estrategia de inteligencia de negocios de las organizaciones. Deben crearse con un objetivo en mente y diseñarse para analizar datos de conjuntos de datos clave a fin de mejorar las decisiones de negocios. En lugar de que los analistas recopilen manualmente las hojas de cálculo, las plataformas de BI modernas pueden acceder a los datos, analizarlos, mostrarlos y compartirlos mediante *dashboards* basados en web. Con herramientas de inteligencia de negocios eficaces y automatizadas, las partes interesadas pueden crear *dashboards* para revisar los datos, extraer conclusiones y tomar decisiones (14).

### **Funcionalidades claves de los dashboards**

Las plataformas de BI modernas ofrecen numerosas funcionalidades clave similares y presentan muchos *dashboards* con ejemplos reales de algunas o todas ellas (14):

- Interfaz personalizable.

- Interactividad.
- Capacidad de obtener datos prácticamente en tiempo real.
- Acceso desde un navegador web.
- Plantillas estándar.
- Capacidad de compartir para impulsar la colaboración.

Todas estas funcionalidades contribuyen al objetivo de los *dashboards* de BI. Además, aportan beneficios generales para los usuarios y la organización.

### **1.1.6. Indicador clave de rendimiento KPI**

Los indicadores clave de rendimiento (KPI) son los indicadores cuantificables críticos (clave) del progreso hacia un resultado previsto. Los KPI proporcionan un enfoque para la mejora estratégica y operativa, crean una base analítica para la toma de decisiones y ayudan a centrar la atención en lo que más importa (15). Según Tableau, los KPIs, como su nombre lo indica, miden el desempeño, y esto puede aplicarse a proyectos, estrategias y acciones. Con estas métricas, el directivo de una empresa accede a información valiosa que le permite tomar decisiones basadas en datos objetivos y claros sobre cómo va el rendimiento de sus proyectos en un tiempo determinado (16). Oracle asegura que los indicadores clave de rendimiento, o KPI, son medidas que definen los objetivos y metas de negocio específicos que suelen acumularse en estrategias organizativas de mayor tamaño que requieren supervisión, mejora y evaluación, así como realizar un seguimiento de estos. Los KPI tienen valores que se pueden medir y que suelen variar con el tiempo, tienen objetivos para determinar una puntuación y el estado de rendimiento, incluyen dimensiones para poder realizar análisis más específicos, y se pueden comparar a lo largo del tiempo para obtener más información sobre las tendencias e identificar los patrones de rendimiento (17).

#### **Características de los KPIs**

- Proporciona evidencia objetiva del progreso hacia el logro de un resultado deseado (15).
- Mide lo que se pretende medir para ayudar a informar una mejor toma de decisiones (15).
- Ofrece una comparación que mida el grado de cambio de rendimiento a lo largo del tiempo (15).

- Puede realizar un seguimiento de la eficiencia, la eficacia, la calidad, la puntualidad, la gobernanza, el cumplimiento, los comportamientos, la economía, el rendimiento del proyecto, el rendimiento del personal o la utilización de los recursos (15).

### **Indicadores tecnológicos**

Para el desarrollo del sistema se tendrá en consideración el uso de indicadores científico-tecnológicos como indicadores KPI. Es decir, se emplearán KPIs tecnológicos. Pero para entender este concepto, es importante conocer qué son los indicadores de ciencia, tecnología e innovación.

Según el Observatorio Tecnológico de la Universidad de Alicante OVTT, los indicadores de ciencia, tecnología e innovación representan instrumentos de medición, análisis y comparación internacional que ayudan a comprender la evolución de las actividades de I+D+i en los sistemas de ciencia, tecnología e innovación, aportando datos estadísticos estratégicos para la inteligencia tecnológica (18).

## **1.2. Sistemas de visualización sobre tecnologías de desarrollo de software**

El siguiente epígrafe trata sobre el análisis de sistemas homólogos. El análisis de sistemas homólogos se usa para comparar diferentes sistemas que tienen una función o estructura similar. Esta técnica se utiliza para identificar similitudes y diferencias entre los sistemas y para identificar las mejores prácticas y soluciones que se pueden aplicar a un sistema específico.

Para llevar a cabo este estudio, se tendrán en cuenta numerosas características a emplear para describir los sistemas homólogos. Estas son:

- Dominio: dominio de aplicación al que pertenece el sistema.
- Tipo de software: privativo, libre o código abierto.
- Enfoque: información que muestra el sistema.
- Metodología: metodología empleada para la búsqueda de información.
- Accesibilidad: determina si existe acceso o no a la información que brinda.
- Autenticación: existencia o no de un sistema de autenticación.
- Sistema de roles: existencia o no de un sistema de roles.

Una vez descritos estos sistemas y explicado su funcionamiento, se procederá a caracterizarlos teniendo en cuenta estos puntos.










### 1.2.1. TIOBE Index

TIOBE está especializada en la evaluación y seguimiento de la calidad del software. Miden la calidad de un sistema de software aplicándole métricas de software ampliamente aceptadas. TIOBE verifica más de millones de líneas de código de software para sus clientes en todo el mundo, en tiempo real, cada día. Miden la calidad del código de software con el objetivo de lograr un mundo libre de errores a través de la experiencia en calidad de software (19).

El índice TIOBE Programming Community es un indicador de la popularidad de los lenguajes de programación. El índice se actualiza una vez al mes. Las clasificaciones se basan en el número de Ingenieros calificados en todo el mundo, cursos y proveedores externos. Motores de búsqueda populares como Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube y Baidu se utilizan para calcular las calificaciones. Es importante tener en cuenta que el índice TIOBE no se trata del mejor lenguaje de programación o el lenguaje en el que se han escrito la mayoría de las líneas de código (19).

El índice se puede utilizar para comprobar si sus habilidades de programación todavía están actualizadas o para hacer una decisión estratégica sobre qué lenguaje de programación debe adoptarse al comenzar a construir un nuevo sistema de software (19).

A continuación, se presenta una imagen con el índice TIOBE del mes de octubre.

Oct 2023	Oct 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.82%	-2.25%
2	2		 C	12.08%	-3.13%
3	4	▲	 C++	10.67%	+0.74%
4	3	▼	 Java	8.92%	-3.92%
5	5		 C#	7.71%	+3.29%
6	7	▲	 JavaScript	2.91%	+0.17%
7	6	▼	 Visual Basic	2.13%	-1.82%
8	9	▲	 PHP	1.90%	-0.14%
9	10	▲	 SQL	1.78%	+0.00%

**Imagen 1.** Ranking de lenguajes de programación Octubre 2023. Fuente: TIOBE Index (2023).



## Definición del Índice de la Comunidad de Programación TIOBE

En las siguientes secciones se explica qué motores de búsqueda califican, qué lenguajes de programación califican y cómo se calculan exactamente las calificaciones.

### Buscadores

Hay 25 motores de búsqueda que se utilizan para calcular el índice TIOBE. Los motores de búsqueda seleccionados son los 25 sitios web mejor clasificados de Similarweb que cumplen con las siguientes condiciones (19):

- La página de entrada del sitio contiene un servicio de búsqueda.
- El resultado de consultar el sitio contiene una indicación del número de visitas a la página.
- Los resultados deben estar disponibles en HTML con etiquetas claras.
- Los motores de búsqueda en idiomas con caracteres especiales deben codificarse correctamente.
- El motor de búsqueda debe devolver al menos 1 *hit* por 1 consulta.
- Los resultados de consultar el sitio no deben contener demasiados valores atípicos.

Sobre la base de estos criterios se califican los siguientes motores de búsqueda (19):

- Google.com: 7,69%
- Wikipedia.org: 7,38%
- Amazon.com: 7,08%
- Vk.com: 6,77%
- Bing.com: 6,46%
- Microsoft.com: 6.15%
- Msn.com: 5,85%
- Ebay.com: 5,54%
- Sharepoint.com: 5,23%
- Google.com.br: 4,92%

- Amazon.co.jp: 4,62%
- Amazon.de: 4,31%
- Rakuten.co.jp: 4,00%
- Etsy.com: 3,69%
- Google.de: 3,38%
- Amazon.co.uk: 3,08%
- Google.co.jp: 2,77%
- Walmart.com: 2,46%
- Amazon.in: 2,15%
- Google.it: 1,85%
- Google.co.uk: 1,54%
- Google.fr: 1.23%
- Google.es: 0,92%
- Ebay.co.uk: 0,62%
- Amazon.ca: 0,31%

## **Lenguaje de programación**

En esta sección se aclara lo que cuenta como lenguaje de programación para el índice TIOBE. Hay 3 requisitos que deben cumplirse (19):

- El lenguaje debe tener una entrada propia en Wikipedia y Wikipedia debe indicar claramente que se trata de un lenguaje de programación. Esta es la razón por la cual (Ruby on) Rails, Excel, Android, Boost, Cocoa, ASP y AJAX no se consideran lenguajes de programación para el índice.
- El lenguaje de programación debe ser Turing completo. Como consecuencia, HTML y XML no se consideran lenguajes de programación.
- El lenguaje de programación debe tener al menos 5.000 visitas para Google.

Los lenguajes de programación que son muy similares se agrupan. Actualmente el máximo

de los aciertos de los lenguajes individuales se tiene en cuenta al calcular las calificaciones de las agrupaciones (19).

La definición de qué lenguajes se agrupan se ha formalizado de acuerdo con las siguientes reglas (19):

- Si un lenguaje tiene su propia entrada de Wikipedia, no se agrupará con otro lenguaje.
- Si un lenguaje A redirige automáticamente a otra entrada de Wikipedia B, A será agrupados junto con B.
- Si un lenguaje A no tiene una entrada separada de Wikipedia, pero se menciona como parte de otra entrada B, A de Wikipedia se agrupará junto con B.

## **Calificaciones**

Las calificaciones se calculan contando los resultados de los motores de búsqueda más populares.

### **1.2.2. PYPL Popularity of Programming Language Index**

El Índice de Popularidad de Lenguajes de Programación de PYPL se crea analizando la frecuencia con la que se buscan tutoriales de lenguajes de programación en Google. Cuanto más se busca un tutorial de lenguajes, más popular se supone que es. Los datos provienen de Google Trends. El índice de popularidad del lenguaje de programación de PYPL puede ayudar a decidir qué lenguaje estudiar o cuál usar en un nuevo proyecto de software (20).

PYPL cuenta, además, con un índice de popularidad dedicado a entornos de desarrollo integrado (IDE) y bases de datos (BD). El índice Top IDE se crea analizando la frecuencia con la que se busca la página de descarga de los IDE en Google, mientras que el Top BD lo hace usando el mismo algoritmo que el índice PYPL anteriormente explicado. Los datos también provienen de Google Trends.

## **Funcionamiento**

Se exportan los datos de Google Trends en formato CSV, y luego se analizan usando Pandas en Python. Primero, se calcula el interés de los tutoriales de cada lenguaje en relación con los tutoriales de Java cada mes. Normalizando el total al 100% se obtiene la cuota de interés en cada

lenguaje, es decir, su popularidad (20). Mismo procedimiento ocurre con los IDE y bases de datos. En el caso de los IDE, se calcula el interés en relación con Visual Studio y las bases de datos en relación con Oracle.

### Diferencias con TIOBE Index

El índice TIOBE cuenta el número de páginas web con el nombre del lenguaje. La programación de Objective-C tiene más de 20 millones de páginas en la web, mientras que la programación en C tiene solo 11 millones. Esto explica por qué Objective-C tiene una alta clasificación TIOBE. Pero, casi nadie lee esas páginas web de Objective-C, según datos de Google Trends. La programación de Objective C se busca 30 veces menos que la programación C (20).

La siguiente figura muestra un ejemplo de este índice.

Worldwide, Oct 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	28.05 %	+0.1 %
2		Java	15.88 %	-1.0 %
3		JavaScript	9.27 %	-0.3 %
4		C#	6.79 %	-0.2 %
5		C/C++	6.59 %	+0.3 %
6		PHP	4.86 %	-0.4 %
7		R	4.45 %	+0.4 %
8		TypeScript	2.93 %	+0.1 %
9	↑↑	Swift	2.69 %	+0.7 %
10	↓	Objective-C	2.29 %	+0.2 %

**Imagen 2.** Ranking de lenguajes de programación Octubre 2023. Fuente: PYPL Index (2023).

## **Resultado del estudio de sistemas homólogos**

TIOBE Index y PYPL constituyen sistemas homólogos de referencia, de los que se han identificado algunas características, categorizadas en ventajas y desventajas.

### **TIOBE Index**

#### **Ventajas:**

- Larga trayectoria: es un índice que se ha mantenido y actualizado regularmente desde 2001, lo que le ha otorgado una larga trayectoria en la medición de la popularidad de los lenguajes de programación.
- Metodología basada en motores de búsqueda: utiliza una metodología que se basa en el recuento de los resultados de búsqueda en varios motores de búsqueda populares. Esto proporciona un enfoque basado en datos para medir la popularidad de los lenguajes.
- Amplia cobertura de lenguajes: cubre una gran cantidad de lenguajes de programación, incluyendo lenguajes populares como Java, C, Python, JavaScript, entre otros. Esto proporciona una visión general amplia de la popularidad relativa de diferentes lenguajes en la industria.

#### **Desventajas:**

- Limitaciones en la metodología: tiene ciertas limitaciones, ya que se basa principalmente en el recuento de resultados de búsqueda en motores de búsqueda. Esto puede llevar a una sobreestimación de la popularidad de ciertos lenguajes que generan más información relacionada en línea.
- Enfoque en términos de búsqueda: se centra en términos de búsqueda en lugar de en el uso real de los lenguajes de programación. Como resultado, la popularidad medida puede estar influenciada por factores como la curiosidad o interés académico en un lenguaje, en lugar de su uso práctico en proyectos y aplicaciones reales.
- Inexistencia de un sistema de autenticación: carece de un sistema de autenticación. Esto producto a que el acceso para visualizar el índice es público.
- Inexistencia de un sistema de roles: carece de un sistema de roles.

- Inaccesibilidad de los datos: debido a su naturaleza privativa, no se tiene acceso a los datos utilizados para calcular su índice.
- Limitaciones de tecnologías de desarrollo de software: solo se limita a representar datos sobre lenguajes de programación, y no de otras tecnologías de desarrollo de software. Por ejemplo: *frameworks*, bases de datos, entornos de desarrollo integrado.
- Escasas visualizaciones: carece de visualizaciones que pueden brindar un mayor nivel de abstracción y análisis hacia los datos.

## **PYPL Index**

### **Ventajas:**

- Enfoque basado en búsquedas de tutoriales: mide la popularidad de los lenguajes de programación mediante el análisis de las búsquedas de tutoriales en los motores de búsqueda. Esto proporciona un indicador de la demanda de aprendizaje y conocimiento de lenguajes de programación en tiempo real.
- Accesibilidad de los datos: debido a su naturaleza de código abierto, publica los datos utilizados para calcular su índice, lo que permite una mayor transparencia y la posibilidad de verificar y analizar los resultados por parte de la comunidad.
- Enfoque en la adopción y el aprendizaje: se enfoca en la popularidad de los lenguajes de programación en términos de su tasa de adopción y demanda de aprendizaje. Esto puede proporcionar información útil para quienes deseen comenzar a aprender o utilizar nuevos lenguajes de programación.

### **Desventajas:**

- Enfoque limitado: utiliza una metodología específica basada en búsquedas de tutoriales, lo que puede limitar su capacidad para representar toda la gama de usos y aplicaciones de los lenguajes de programación.
- Posible sesgo en los datos: la popularidad medida por el PYPL Index puede verse afectada por sesgos en las búsquedas de tutoriales en línea. Esto puede influir en las clasificaciones y no necesariamente reflejar a cabalidad la popularidad real de los lenguajes.

- Escasa representación de lenguajes de nicho: tiende a enfocarse en los lenguajes de programación más populares y utilizados ampliamente. Esto puede resultar en una escasa representación de lenguajes más especializados o de nicho que no generen una demanda significativa de aprendizaje en línea.
- Inexistencia de un sistema de autenticación: carece de un sistema de autenticación. Esto producto a que el acceso para visualizar el índice es público.
- Inexistencia de un sistema de roles: carece de un sistema de roles.
- Escasas visualizaciones: carece de visualizaciones que pueden brindar un mayor nivel de abstracción y análisis hacia los datos.

La siguiente tabla resume las ventajas y desventajas de estos sistemas teniendo en cuenta las características definidas al inicio del epígrafe.

<b>Características</b>	<b>TIOBE</b>	<b>PYPL</b>
<b>Dominio</b>	Webapp	Webapp
<b>Tipo de software</b>	Privativo	Código abierto
<b>Enfoque</b>	Muestra información sobre la popularidad de lenguajes de programación.	Muestra información sobre la popularidad de lenguajes de programación, bases de datos y entornos de desarrollo integrado.
<b>Metodología</b>	Se basa en la recopilación de información a partir de resultados en varios motores de búsquedas. Emplea escasas visualizaciones (tabla, gráfico de líneas).	Se basa en la recopilación de información a partir de resultados en Google. Emplea escasas visualizaciones (tabla, gráfico de líneas).
<b>Accesibilidad</b>	No se puede acceder a sus datos.	Se puede acceder a sus datos y se permite su descarga.
<b>Autenticación</b>	No posee	No posee

<b>Sistema de roles</b>	No posee	No posee
-------------------------	----------	----------

**Tabla 1.** Características de los sistemas homólogos. Elaboración propia.

### **Conclusiones del estudio de sistemas homólogos**

TIOBE Index y PYPL constituyen sistemas homólogos de referencia, de los que se han extraído funcionalidades para su implementación en el sistema de visualización de indicadores. Si bien servirán como fuentes importantes de datos, estos carecen de ciertas características que son necesarias incluir. Por lo tanto, se propone desarrollar un sistema completamente nuevo, con nuevas funcionalidades, pero que mantenga la esencia de estos sistemas. La idea recae en subsanar todas aquellas deficiencias que tienen estos sistemas. Por lo que se propone incluir las siguientes características en el sistema de visualización de indicadores.

### **Características del sistema de visualización de indicadores**

- Enfoque menos limitado: utiliza las metodologías empleadas por TIOBE y PYPL, y un nuevo ranking a partir del uso de nuevos indicadores. Empleará 3 rankings.
- Existencia de un sistema de autenticación: posee un sistema de autenticación. Esto producto a que el acceso para visualizar los datos está limitado a un grupo específico de profesionales.
- Existencia de un sistema de roles: posee un sistema de roles. Esto se debe a que los usuarios del sistema tendrán permisos delimitados sobre el mismo.
- Amplia representación de tecnologías de desarrollo de software: tiende a enfocarse en muchos lenguajes de programación y otras tecnologías de desarrollo de software.
- Variedad de visualizaciones: posee amplia variedad de visualizaciones que pueden brindar un mayor nivel de abstracción y análisis hacia los datos. El usuario tiene libertad de manipulación y configuración sobre las visualizaciones.
- Accesibilidad de los datos: debido a su naturaleza de código abierto, se tiene acceso a los datos utilizados para calcular su índice. Los usuarios serán capaces de descargar los datos y generar imágenes de las visualizaciones representadas.

La próxima tabla resume las características que poseerá la propuesta de solución, teniendo en cuenta los puntos definidos al inicio del epígrafe y utilizados en la caracterización de los



sistemas homólogos.

<b>Características</b>	<b>Sistema de visualización de indicadores</b>
<b>Dominio</b>	Webapp
<b>Tipo de software</b>	Privativo
<b>Enfoque</b>	Muestra información sobre la popularidad de tecnologías de desarrollo de software como lenguajes de programación, <i>frameworks</i> , bases de datos, entornos de desarrollo integrado, sistemas operativos, aplicaciones.
<b>Metodología</b>	Se basa en la recopilación de información a partir de resultados en varios motores de búsquedas y fuentes de información. Emplea múltiples visualizaciones (tabla, gráfico de líneas, gráfico de barras, gráfico de puntos, gráfico circular).
<b>Accesibilidad</b>	Se puede acceder a sus datos y permite su descarga. Permite renderizar imágenes de los gráficos.
<b>Autenticación</b>	Posee
<b>Sistema de roles</b>	Posee

**Tabla 2.** Características del sistema de visualización de indicadores. Elaboración propia.

### 1.3. Metodología de desarrollo de software

En la década de 1990 el descontento con los enfoques engorrosos de la ingeniería de software condujo a algunos desarrolladores de software a proponer nuevos métodos ágiles, los cuales permitieron que el equipo de desarrollo se enfocara en el software en lugar del diseño y la documentación. Los métodos ágiles se apoyan universalmente en el enfoque incremental para la especificación, el desarrollo y la entrega del software. Son más adecuados para el diseño de aplicaciones en que los requerimientos del sistema cambian, por lo general, rápidamente durante el proceso de desarrollo. Tienen la intención de entregar con prontitud el software operativo a los clientes, quienes entonces propondrán requerimientos nuevos y variados para incluir en posteriores iteraciones del sistema. Se dirigen a simplificar el proceso burocrático al evitar trabajo

con valor dudoso a largo plazo, y a eliminar documentación que quizá nunca se emplee (21).

## Programación Extrema XP

La programación extrema (XP) es quizás el método ágil mejor conocido y más ampliamente usado. El nombre fue acuñado debido a que el enfoque se desarrolló llevando a niveles “extremos” las prácticas reconocidas, como el desarrollo iterativo. Por ejemplo, en la XP muchas versiones actuales de un sistema pueden desarrollarse mediante diferentes programadores, integrarse y ponerse a prueba en un solo día (21).

La programación extrema incluye algunas prácticas, resumidas en la siguiente tabla, las cuales reflejan los principios de los métodos ágiles.

Principio	Descripción
Planeación incremental	Los requerimientos se registran en tarjetas de historia ( <i>story cards</i> ) y las historias que se van a incluir en una liberación se determinan por el tiempo disponible y la prioridad relativa. Los desarrolladores desglosan dichas historias en “tareas” de desarrollo.
Liberaciones pequeñas	Al principio se desarrolla el conjunto mínimo de funcionalidad útil, que ofrece valor para el negocio. Las liberaciones del sistema son frecuentes y agregan incrementalmente funcionalidad a la primera liberación.
Diseño simple	Se realiza un diseño suficiente para cubrir sólo aquellos requerimientos actuales.
Desarrollo de la primera prueba	Se usa un marco de referencia de prueba de unidad automatizada al escribir las pruebas para una nueva pieza de funcionalidad, antes de que esta última se implemente.
Refactorización	Se espera que todos los desarrolladores refactoricen de manera continua el código y, tan pronto como sea posible, se encuentren mejoras de éste. Lo anterior

	conserva el código simple y mantenible.
Programación en pares	Los desarrolladores trabajan en pares, y cada uno comprueba el trabajo del otro; además, ofrecen apoyo para que se realice siempre un buen trabajo.
Propiedad colectiva	Los desarrolladores en pares laboran en todas las áreas del sistema, de manera que no se desarrollan islas de experiencia, ya que todos los desarrolladores se responsabilizan por todo el código. Cualquiera puede cambiar cualquier función.
Integración continua	Tan pronto como esté completa una tarea, se integra en todo el sistema. Después de tal integración, deben aprobarse todas las pruebas de unidad en el sistema.
Ritmo sustentable	Grandes cantidades de tiempo extra no se consideran aceptables, pues el efecto neto de este tiempo libre con frecuencia es reducir la calidad del código y la productividad de término medio.
Cliente en sitio	Un representante del usuario final del sistema (el cliente) tiene que disponer de tiempo completo para formar parte del equipo XP. En un proceso de programación extrema, el cliente es miembro del equipo de desarrollo y responsable de llevar los requerimientos del sistema al grupo para su implementación.

**Tabla 3.** Principios de la metodología XP. Fuente: Sommerville (2011).

Se elige XP para el desarrollo de la investigación ya que constituye una de las metodologías de desarrollo ágil más usadas en la ingeniería de software. Además, se emplea en la Universidad de Ciencias Informáticas con gran frecuencia. Teniendo en cuenta que este proyecto constituye solamente un módulo de un ecosistema, se elige emplear XP para mantener constancia de uso de esta metodología, que también se emplea en el resto de proyectos que conforman el ecosistema, y de esta manera apuntalar a una única forma de desarrollo. XP promueve una

actitud flexible hacia los cambios y se basa en la premisa de que los requisitos y circunstancias cambian con el tiempo, por lo que la metodología permite adaptarse rápidamente a estos cambios. Esto se logra a través de prácticas como la planificación ágil, las historias de usuario y el enfoque en iteraciones cortas. XP se basa en iteraciones cortas y entregas frecuentes de software. Esto permite obtener valor y funcionalidad desde las primeras etapas del proyecto. La entrega temprana también permite obtener retroalimentación rápida y mejorar el producto a medida que avanza el desarrollo. XP no impone un conjunto específico de herramientas o prácticas, lo que brinda flexibilidad para adaptar y elegir las herramientas y prácticas que mejor se adapten a las necesidades y contexto.

## **1.4. Herramientas y tecnologías empleadas en el desarrollo de la propuesta de solución**

### **1.4.1. Herramienta de visualización**

El software de código abierto de Grafana permite consultar, visualizar, alertar y explorar sus métricas, registros y rastros dondequiera que estén almacenados. Grafana OSS proporciona herramientas para convertir datos de base de datos de series temporales (TSDB) en gráficos y visualizaciones perspicas. El marco de complementos Grafana OSS también permite conectar otras fuentes de datos como bases de datos NoSQL / SQL, herramientas de tickets como Jira o ServiceNow, y herramientas de CI / CD como GitLab (22).

Las siguientes secciones proporcionan una descripción general de las características de Grafana.

### **Explorar métricas, registros y seguimientos**

Grafana permite explorar datos a través de consultas y desglose dinámico. Divide la vista y compara diferentes rangos de tiempo, consultas y fuentes de datos en paralelo (22).

### **Panel de control**

Un panel de control de Grafana consiste en paneles que muestran datos en gráficos, tablas y otras visualizaciones. Estos paneles se crean utilizando componentes que transforman los datos sin procesar de una fuente de datos en visualizaciones. El proceso implica pasar datos a través

de tres puertas: un complemento, una consulta y una transformación opcional (22).



**Imagen 3.** Ejemplo de panel de control en Grafana. Fuente: Grafana Documentation (2023).

## Fuentes de datos

Un origen de datos hace referencia a cualquier entidad que consta de datos. Puede ser una base de datos SQL, Grafana Loki, Grafana Mimir o una API basada en JSON. Incluso puede ser un archivo CSV básico. El primer paso para crear una visualización de panel es seleccionar el origen de datos que contiene los datos que necesita (22).

## **Plugins**

Un *plugin* de Grafana es un software que añade nuevas capacidades a Grafana. Vienen en muchos tipos, como paneles y fuentes de datos.

## **Paneles**

Después de que los datos se obtienen, consultan y transforman, pasan a un panel, que es la puerta final en el viaje hacia una visualización de Grafana. Un panel es un contenedor que muestra la visualización y proporciona varios controles para manipularla (22).

## **Variables del panel**

Las variables de plantilla permiten crear paneles que se pueden reutilizar para muchos casos de uso diferentes. Los valores no están codificados de forma rígida con estas plantillas, por lo que, por ejemplo, si tiene un servidor de producción y un servidor de prueba, puede usar el mismo panel para ambos (22).

## **Importar/exportar paneles**

Grafana ofrece múltiples formas de importar y exportar paneles, ya sea desde la interfaz de usuario, utilizando archivos JSON o a través de la API. Esta flexibilidad permite compartir configuraciones de paneles de manera eficiente entre diferentes entornos y colaboradores.

## **Autenticación**

La autenticación en Grafana se puede lograr a través de varios métodos, como nombre de usuario y contraseña, OAuth, LDAP, Auth Proxy y tokens de autenticación. La elección del método adecuado depende de las necesidades de seguridad y las preferencias de la organización.

## **Permisos**

Permite establecer privilegios sobre paneles y usuarios. Estos privilegios establecen los permisos que tienen los usuarios sobre los paneles de datos y Grafana en general. La herramienta categoriza a los usuarios en tres roles: visualizador, editor y administrador. Cada uno

de ellos tiene permisos predeterminados sobre Grafana.

Se elige Grafana como herramienta de visualización ya que tiene varias ventajas y aporta beneficios al proyecto. Entre estos beneficios se pueden destacar:

- Flexibilidad y variedad de visualizaciones: Grafana ofrece una amplia gama de opciones de visualización, desde gráficos de líneas y barras hasta mapas, tablas y medidores. Además, permite la personalización y configuración avanzada de las visualizaciones, lo que te brinda flexibilidad para representar los datos de la manera más adecuada para tus necesidades específicas.
- Fácil integración con diversas fuentes de datos: Grafana tiene una extensa lista de conectores que permiten la integración con muchas fuentes de datos populares. Esto permite obtener datos en tiempo real y visualizarlos en una interfaz centralizada sin tener que cambiar de herramientas o consultar múltiples sistemas.
- Interactividad y exploración de datos: una de las fortalezas de Grafana es su capacidad para proporcionar interactividad en las visualizaciones. Permite a los usuarios explorar y analizar los datos mediante la selección y el filtrado de series, el zoom, la comparación y mucho más. Esto facilita la comprensión y el análisis profundo de los datos visualizados.
- Paneles de control personalizables: Grafana permite crear y personalizar paneles de control a medida según las necesidades y objetivos. Puedes combinar múltiples visualizaciones en un único panel, organizarlos en diseños personalizados, agregar anotaciones, alertas y hacer uso de variables para facilitar la interactividad y la adaptación a diferentes escenarios.
- Comunidad activa y ecosistema de complementos: Grafana cuenta con una comunidad activa que desarrolla y comparte complementos y extensiones. Esto amplía aún más las capacidades de Grafana y brinda acceso a una amplia variedad de complementos y plantillas predefinidas que puedes utilizar para enriquecer las visualizaciones y personalizarlas de acuerdo a tus necesidades específicas.
- Código abierto y gratuito: Grafana es una herramienta de software de código abierto distribuida bajo la licencia Apache 2.0. Esto significa que puede utilizarse de forma gratuita y tener acceso al código fuente, lo que da la libertad de modificar y personalizar la herramienta según las necesidades.

- Webapp: Grafana pertenece al dominio de aplicaciones web, por lo que se puede acceder a esta herramienta desde una PC conectada a la red a través de un navegador. Se garantiza usar aplicación web ya que el resto de módulos que pertenecen al ecosistema del proyecto también pertenecen al dominio de *webapp*.

#### **1.4.2. Herramienta CASE**

Visual Paradigm es una suite de desarrollo de software y gestión empresarial galardonada líder en todo el mundo, que proporciona todas las características que necesita para la arquitectura empresarial, la gestión de proyectos, el desarrollo de software y la colaboración en equipo en una solución de ventanilla única. Visual Paradigm permite a su equipo gestionar la complejidad de la transformación empresarial para hacer frente a los mercados, las tecnologías y los requisitos normativos que cambian rápidamente. Es una solución integral ideal para la planificación de la arquitectura empresarial y la transformación empresarial, la gestión de proyectos y el desarrollo ágil de software para que su empresa pueda mantener el control y fomentar el crecimiento (23).

#### **1.4.3. Lenguaje de modelado**

El lenguaje de modelado unificado es un estándar de la industria que fue creado por el Grupo de gestión de objetos OMG para modelar y documentar el proceso de desarrollo de sistemas orientados a objetos. Proporciona un marco para que el proceso de desarrollo incluya todo, desde las especificaciones de los requisitos, pasando por el diseño arquitectónico, hasta la codificación y documentación de las pruebas (24).

Esta norma define reglas y notaciones para especificar sistemas empresariales y de software. La notación proporciona un rico conjunto de elementos gráficos para modelar sistemas orientados a objetos, y las reglas establecen cómo se pueden conectar y usar esos elementos. UML no es una herramienta para crear sistemas de software. En cambio, es un lenguaje visual para comunicar, modelar, especificar y definir sistemas (24).

Este lenguaje está diseñado para ser flexible, extensible y completo, pero lo suficientemente genérico como para servir como base para todas las necesidades de modelado de sistemas. Con su especificación, hay una amplia gama de elementos caracterizados por los tipos de diagramas que sirven y los atributos que demostraron. Todo se puede especificar aún más mediante el uso de estereotipos, etiquetas y perfiles (24).



#### **1.4.4. Entorno de desarrollo integrado**

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C ++, C #, Java, Python, PHP, Go, .NET) (25).

#### **Multiplataforma**

Visual Studio Code es extremadamente versátil y compatible con Windows, macOS y Linux, lo que lo convierte en una excelente opción para desarrolladores que buscan un editor de código potente y multiplataforma.

#### **Extensiones**

Las extensiones de VS Code permiten a terceros agregar soporte para (25):

- Lenguajes - C++, C#, Go, Java, Python.
- Herramientas - ESLint, JSHint, PowerShell.
- Depuradores - PHP XDebug.
- Mapas claves - Vim, Sublime Text, IntelliJ, Emacs, Atom, Brackets, Visual Studio, Eclipse.

Las extensiones se integran en la interfaz de usuario, los comandos y los sistemas de ejecución de tareas de VS Code, por lo que le resultará fácil trabajar con diferentes tecnologías a través de la interfaz compartida de VS Code (25).

#### **Componentes y herramientas adicionales**

Visual Studio Code es una pequeña descarga por diseño y solo incluye el número mínimo de componentes compartidos en la mayoría de los flujos de trabajo de desarrollo. Se incluyen funciones básicas como el editor, la administración de archivos, la administración de ventanas y la configuración de preferencias. Un servicio de lenguaje JavaScript/TypeScript y el depurador Node.js también forman parte de la instalación base. La mayoría de los usuarios de VS Code necesitarán instalar componentes adicionales dependiendo de sus necesidades específicas (25).

Se elige Visual Studio Code como entorno de desarrollo ya que aporta varios beneficios y ventajas. Algunas de estas ventajas pueden ser:

- **Ligereza y rendimiento:** Visual Studio Code es conocido por su ligereza y rendimiento. Aunque es un editor de código potente, consume menos recursos en comparación con otras herramientas de desarrollo de mayor tamaño y complejidad. Esto permite una experiencia de desarrollo más rápida y sin retrasos, incluso en sistemas más modestos.
- **Multiplataforma y compatible con múltiples lenguajes:** Visual Studio Code es compatible con múltiples sistemas operativos, incluyendo Windows, macOS y Linux. Además, cuenta con soporte para una amplia variedad de lenguajes de programación, incluyendo JavaScript, Python, Java, C#, HTML/CSS y muchos más.
- **Amplia extensibilidad y ecosistema:** Visual Studio Code cuenta con una gran cantidad de extensiones disponibles, lo que permite personalizar y ampliar el editor según las necesidades específicas.
- **Potentes capacidades de edición y productividad:** Visual Studio Code ofrece varias características y funcionalidades que mejoran la productividad del desarrollador. Esto incluye resaltado de sintaxis, autocompletado inteligente, sugerencias contextuales, refactorización de código, integración con control de versiones, navegación rápida por archivos y una amplia gama de atajos de teclado personalizables. Todo esto ayuda a agilizar el flujo de trabajo y a escribir código más eficiente y libre de errores.
- **Integración con servicios y herramientas populares:** Visual Studio Code se integra con muchos servicios y herramientas populares utilizadas en el desarrollo de software, como Git, Azure, Docker, Kubernetes y más. Esto simplifica la configuración y el uso de estos servicios directamente desde el editor, lo que facilita el desarrollo y el despliegue de aplicaciones en diferentes entornos.

#### **1.4.5. Lenguaje de programación**

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas

áreas, para la mayoría de plataformas. Python está disponible en los sistemas operativos Windows, macOS y Unix (26).

## **Python para análisis de datos**

Python ha desarrollado una comunidad de análisis de datos y computación científica muy grande y activa. En los últimos 20 años, Python ha pasado de ser un lenguaje de ciencia computacional de vanguardia a uno de los lenguajes más importantes para la ciencia de datos, el aprendizaje automático y el desarrollo general de software, tanto académicamente hablando como dentro del sector (27).

Para análisis de datos, computación interactiva y visualización de datos, es inevitable que Python dé lugar a comparaciones con otros lenguajes de programación y herramientas de fuente abierta y comerciales de uso generalizado, como R, MATLAB, SAS, Stata. En los últimos años, las librerías de código abierto mejoradas de Python (como pandas y scikitlearn) lo han convertido en la opción habitual para tareas de análisis de datos. Combinadas con la solidez global de Python para ingeniería de software genérica, es una excelente alternativa como lenguaje principal para crear aplicaciones de datos (27).

## **Pandas**

Pandas es una biblioteca de código abierto, con licencia BSD que proporciona un alto rendimiento, estructuras de datos fáciles de usar y herramientas de análisis de datos para el lenguaje de programación Python (28).

### **Aspectos destacados de la biblioteca**

- Un objeto *DataFrame* rápido y eficiente para la manipulación de datos con indexación integrada (29).
- Herramientas para leer y escribir datos entre estructuras de datos en memoria y diferentes formatos: CSV y archivos de texto, Microsoft Excel, bases de datos SQL, y el rápido formato HDF5 (29).
- Alineación inteligente de datos y manejo integrado de datos faltantes: Obtenga alineación automática basada en etiquetas en cálculos y manipule fácilmente datos desordenados

en una forma ordenada (29).

- Remodelación flexible y pivote de conjuntos de datos (29).
- Segmentación inteligente basada en etiquetas, indexación sofisticada y subconjunto de grandes conjuntos de datos (29).
- Las columnas se pueden insertar y eliminar de las estructuras de datos por tamaño mutabilidad (29).
- Agregación o transformación de datos con un potente grupo por motor permitir operaciones de división-aplicación-combinación en conjuntos de datos (29).
- Fusión y unión de conjuntos de datos de alto rendimiento (29).
- La indexación de ejes jerárquicos proporciona una forma intuitiva de trabajar con datos de alta dimensión en una estructura de datos de dimensión inferior (29).
- Funcionalidad de series temporales: generación de intervalos de fechas y frecuencia conversión, mover estadísticas de ventanas, cambio de fecha y retraso. Incluso crear compensaciones de tiempo específicas del dominio y tiempo de unión series sin perder datos (29).
- Altamente optimizado para el rendimiento, con rutas de código críticas escritas en Cython o C (29).
- Python con pandas está en uso en una amplia variedad de dominios comerciales, incluyendo Finanzas, Neurociencia, Economía, Estadísticas, publicidad, analítica web y más (29).

#### **1.4.6. Gestor de base de datos**

MongoDB es una base de datos NoSQL documental y de código abierto que proporciona alto rendimiento, alta disponibilidad y escalabilidad automática. Un registro en MongoDB es un documento, que es una estructura de datos compuesta por pares de campos y valores. Los documentos en MongoDB son similares a los objetos JSON. Los valores de los campos pueden incluir otros documentos, matrices y matrices de documentos (30).

Las ventajas de utilizar documentos son (30):

- Los documentos (es decir, los objetos) corresponden a tipos de datos nativos en muchos

lenguajes de programación.

- Los documentos y matrices integrados reducen la necesidad de uniones costosas.
- El esquema dinámico admite el polimorfismo fluido.

## **Colecciones**

MongoDB almacena documentos en colecciones. Las colecciones son análogas a las tablas de las bases de datos relacionales (31).

## **Características principales**

### **Alto rendimiento**

MongoDB proporciona persistencia de datos de alto rendimiento. En particular (31):

- La compatibilidad con modelos de datos integrados reduce la actividad de E/S en la base de datos.
- Los índices admiten consultas más rápidas y pueden incluir claves de incrustaciones incrustadas documentos y matrices.

### **API de consulta**

La API de consulta de MongoDB admite leer y escribir operaciones (CRUD) así como (31):

- Agregación de datos
- Búsqueda de texto y Consultas geoespaciales.

### **Alta disponibilidad**

La instalación de replicación de MongoDB, llamada Conjunto de réplicas, proporciona (31):

- Conmutación por error automática.
- Redundancia de datos.
- Un Conjunto de réplicas es un grupo de Servidores MongoDB que mantienen el mismo conjunto de datos, proporcionando redundancia y aumentar la disponibilidad de datos.

## **Escalabilidad horizontal**

MongoDB proporciona escalabilidad horizontal como parte de su funcionalidad principal (31). Se refiere a su capacidad para manejar un aumento en la carga de trabajo o la demanda agregando más recursos en forma horizontal, es decir, mediante la adición de más instancias (o nodos) que funcionan en paralelo.

Se elige MongoDB como sistema de gestión de bases de datos por los beneficios que este brinda. A continuación, se mencionarán algunos de estos beneficios:

- **Modelo de datos flexible:** MongoDB utiliza un modelo de datos flexible denominado documento JSON (BSON), en lugar del modelo de tablas utilizado en las bases de datos relacionales. Esto permite un almacenamiento de datos más flexible y adaptable, ya que los documentos pueden tener diferentes estructuras y esquemas.
- **Escalabilidad horizontal:** MongoDB está diseñado para escalar horizontalmente, lo que significa que puede distribuir los datos en múltiples servidores o clústeres. Esto permite manejar grandes volúmenes de datos y cargas de trabajo intensivas a medida que crece la aplicación.
- **Alta disponibilidad y tolerancia a fallos:** MongoDB ofrece opciones para la replicación y tolerancia a fallos. Puede configurar conjuntos de réplicas, que consisten en varios nodos de MongoDB que sincronizan automáticamente los datos entre sí. Esto proporciona redundancia y alta disponibilidad en caso de fallos, ya que los datos se replican en varios servidores y se puede lograr una conmutación por error rápida y sin interrupciones.

### **1.4.7. Herramientas de pruebas de software**

#### **OSWAP ZAP**

Zed Attack Proxy (ZAP) es una herramienta de pruebas de penetración gratuita y de código abierto que se mantiene bajo el paraguas de El Proyecto de Seguridad de Software (SSP). ZAP está diseñado específicamente para probar aplicaciones web y es flexible y extensible (32).

En esencia, ZAP es lo que se conoce como un "proxy de intermediario". Se encuentra entre el navegador del evaluador y la aplicación web para que pueda interceptar e inspeccionar los

mensajes enviados entre el navegador y la aplicación web, modificar el contenido si es necesario y, a continuación, reenviar esos paquetes al destino. Se puede utilizar como una aplicación independiente (32).

ZAP proporciona funcionalidad para una variedad de niveles de habilidad, desde desarrolladores hasta probadores nuevos en pruebas de seguridad y especialistas en pruebas de seguridad. ZAP tiene versiones para cada sistema operativo y Docker principales, por lo que no está atado a un solo sistema operativo. La funcionalidad adicional está disponible gratuitamente en una variedad de complementos en ZAP Marketplace, a los que se puede acceder desde el cliente ZAP (32).

Debido a que ZAP es de código abierto, el código fuente se puede examinar para ver exactamente cómo se implementa la funcionalidad. Cualquiera puede ofrecerse como voluntario para trabajar en ZAP, corregir errores, agregar características, crear solicitudes de incorporación de cambios para incorporar correcciones al proyecto y crear complementos para admitir situaciones especializadas (32).

## **Grafana k6**

Grafana k6 es una herramienta de prueba de carga de código abierto que hace que las pruebas de rendimiento sean fáciles y productivas para los equipos de ingeniería. K6 es gratuito, está centrado en el desarrollador y es extensible (33).

Con k6, puede probar la fiabilidad y el rendimiento de sus sistemas y detectar antes las regresiones y los problemas de rendimiento (33).

K6 es desarrollado por Grafana Labs y su comunidad.

## **Características principales**

Las características clave incluyen:

- Herramienta CLI con API fáciles de usar para desarrolladores (33).
- Scripting en JavaScript ES2015/ES6 con soporte para Módulos locales y remotos (33).
- Cheques y umbrales para pruebas de carga orientadas a objetivos y fáciles de automatizar (33).

## Casos de uso

Los usuarios de k6 suelen ser desarrolladores, ingenieros de control de calidad, SDET y SRE. Usan k6 para probar el rendimiento y la confiabilidad de las API, los microservicios y los sitios web. Los casos de uso de k6 son (33):

- Pruebas de carga: k6 está optimizado para un consumo mínimo de recursos y diseñado para ejecutar pruebas de alta carga (estrés) (33).
- Pruebas de navegador: a través de Navegador K6, puede ejecutar pruebas de rendimiento basadas en el navegador y detectar problemas relacionados solo con los navegadores que se pueden omitir por completo desde el nivel de protocolo (33).
- Monitorización sintética y de rendimiento: con k6, puede automatizar y programar para activar pruebas con mucha frecuencia con una pequeña carga para validar continuamente el rendimiento y la disponibilidad de su entorno de producción (33).

## Postman

Postman es una plataforma de API para crear y usar API. Postman simplifica cada paso del ciclo de vida de la API y agiliza la colaboración para que pueda crear mejores API más rápido (34).

El cliente de API de Postman es la herramienta fundamental de Postman y le permite explorar, depurar y probar fácilmente las API, a la vez que le permite definir solicitudes de API complejas para HTTP, REST, SOAP, GraphQL y WebSockets (34).

El cliente de la API detecta automáticamente el idioma de la respuesta, los enlaces y el texto de formato dentro del cuerpo para facilitar la inspección. El cliente también incluye soporte integrado para protocolos de autenticación como OAuth 1.2/2.0, AWS Signature, Hawk y muchos más (34).

A través del cliente de API, puede organizar las solicitudes en colecciones de Postman para ayudarle a organizar sus solicitudes de reutilización para que no pierda tiempo creando todo desde cero. Sus colecciones también pueden contener código JavaScript para vincular solicitudes o automatizar flujos de trabajo comunes, y puede usar secuencias de comandos para visualizar las respuestas de la API como tablas y gráficos (34).



#### 1.4.8. Fuentes de información

El sistema se nutre de la información obtenida a partir de la recopilación y análisis de múltiples fuentes de información, algunas de las cuales se enumeran a continuación.

La gran mayoría de la información procede de otros sitios web y proyectos, incluidos:

- Github: [github.com](https://github.com)
- Gitlab: [gitlab.com](https://gitlab.com)
- Stackoverflow Insights: [insights.stackoverflow.com](https://insights.stackoverflow.com)
- AWS Amazon: [aws.amazon.com](https://aws.amazon.com)
- Google Cloud: [cloud.google.com](https://cloud.google.com)
- Jupyter: [jupyter.org](https://jupyter.org)
- Linkedin: [linkedin.com](https://linkedin.com)
- Reddit: [reddit.com](https://reddit.com)
- PYPL PopularitY of Programming Language: [pypl.github.io](https://pypl.github.io)
- TIOBE Index: [tiobe.com](https://tiobe.com)
- Twitter: [twitter.com](https://twitter.com)
- Ubuntu: [ubuntu.com](https://ubuntu.com)
- Wikipedia: [wikipedia.org](https://wikipedia.org)

#### Conclusiones parciales

En este capítulo se expusieron los referentes fundamentales y conceptos del trabajo investigativo. Se trataron los conceptos asociados al tema de investigación, y se realizó un estudio de homólogos, donde se caracterizaron otros sistemas similares a la solución mediante el análisis del mercado para analizar su factibilidad. De esta manera, se determinan las características principales que debe comprender el sistema de visualización. Se definieron las metodologías, herramientas y tecnologías para el desarrollo de la solución informática. El marco de trabajo ingenieril para el proceso de desarrollo justifica el uso de la metodología XP. Grafana se emplea como herramienta de visualización para la información que sea capaz de extraerse de las distintas fuentes de datos identificadas. Se utiliza Visual Studio Code como editor de código y Python como

lenguaje de programación, debido a su gran alcance y desempeño en análisis de datos. Visual Paradigm bajo lenguaje UML se usará en la elaboración de artefactos ingenieriles.

# Capítulo II. Análisis y diseño de un sistema de visualización de indicadores para la vigilancia tecnológica

## Introducción al capítulo

El proceso de análisis y diseño de un software es el conjunto de actividades que se realizan para comprender los requisitos del sistema, definir su estructura y funcionalidad, y planificar su implementación. De esta manera, se presenta el proceso de análisis y diseño del sistema de solución informática. Este capítulo incluye: descripción del sistema de visualización, estilo arquitectónico, requisitos informacionales, funcionales, no funcionales, historias de usuario.

### 2.1. Descripción de la propuesta de solución

La propuesta de solución se basa en el dominio de un software de aplicación web, también conocido como *webapp*. El propósito de este sistema de visualización de indicadores para la vigilancia tecnológica es posibilitar a los investigadores el análisis de datos, de forma gráfica, sobre métricas asociadas a temas de investigación y les permita analizar tendencias, establecer estrategias y tomar decisiones. El sistema será implementado usando Grafana, una herramienta de monitoreo para la visualización de datos. Se emplean KPIs científico-tecnológicos para medir el comportamiento de tecnologías de desarrollo de software, como lenguajes de programación y bases de datos, con una frecuencia mensual. El sistema se nutre de múltiples fuentes de información provenientes de la web. Los datos extraídos de estas fuentes se procesan y posteriormente se almacenan para ser visualizados en la herramienta. Basa sus visualizaciones en sistemas de ranking, donde los índices son calculados a partir de los datos almacenados. Cuenta con un sistema de autenticación y roles, que definen el acceso y permisos de los usuarios sobre el sistema. Emplea arquitectura cliente-servidor y usa MongoDB como gestor de base de datos.

#### 2.1.1. Arquitectura cliente-servidor

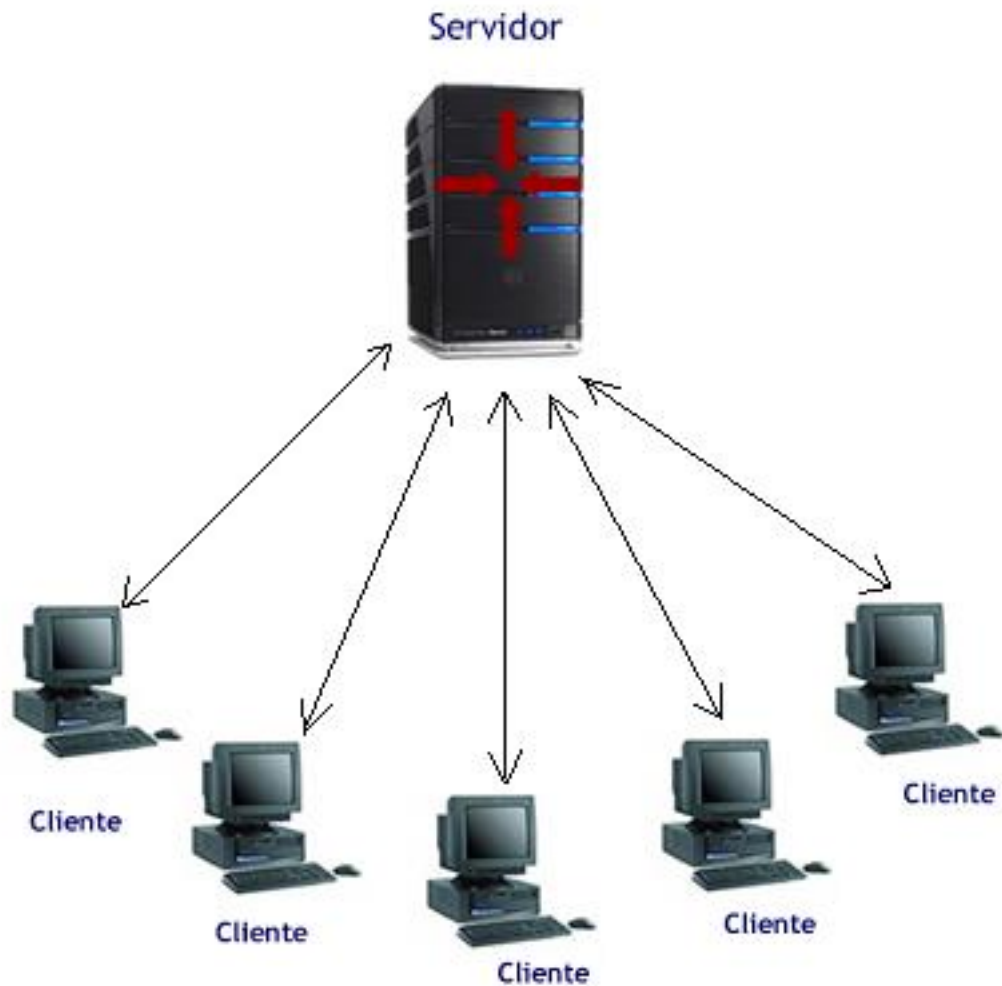
En una arquitectura cliente-servidor, la funcionalidad del sistema se organiza en servicios, y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios y para utilizarlos ingresan a los servidores. Se usa cuando, desde varias ubicaciones, se tiene

que ingresar a los datos en una base de datos compartida. Como los servidores se pueden replicar, también se usan cuando la carga de un sistema es variable (21).

Un sistema que sigue el patrón cliente-servidor se organiza como un conjunto de servicios y servidores asociados, y de clientes que acceden y usan los servicios. Los principales componentes de este modelo son:

- Un conjunto de servidores que ofrecen servicios a otros componentes. Ejemplos de éstos incluyen servidores de impresión; servidores de archivo que brindan servicios de administración de archivos, y un servidor compilador, que proporciona servicios de compilación de lenguaje de programación (21).
- Un conjunto de clientes que solicitan los servicios que ofrecen los servidores. Habrá usualmente varias instancias de un programa cliente que se ejecuten de manera concurrente en diferentes computadoras (21).
- Una red que permite a los clientes acceder a dichos servicios. La mayoría de los sistemas cliente-servidor se implementan como sistemas distribuidos, conectados mediante protocolos de Internet (21).

A continuación, se representa una imagen que detalla el comportamiento de esta arquitectura.

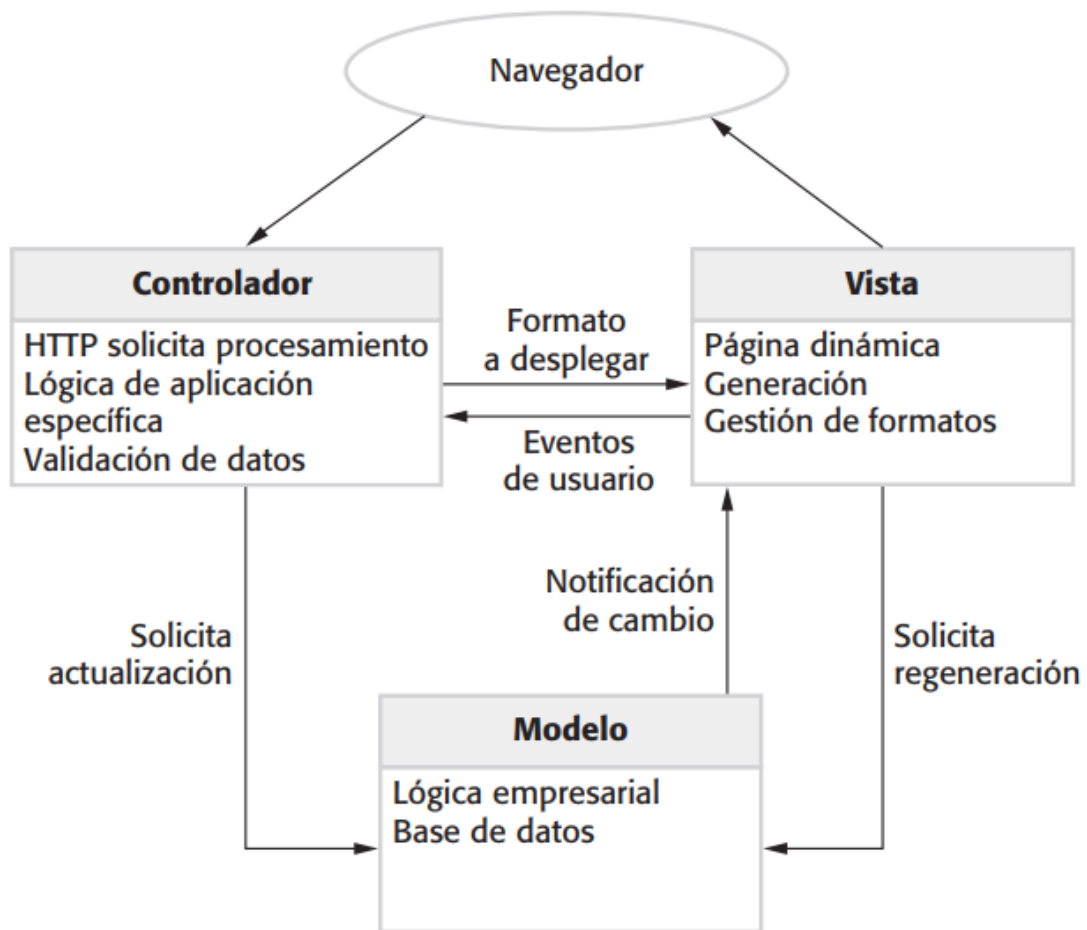


**Imagen 4.** Arquitectura cliente-servidor de la propuesta de solución. Tomado de Google Images.

### 2.1.2. Patrón arquitectónico

El patrón de arquitectura del controlador de vista de modelos (MVC) separa una aplicación en tres grupos de componentes principales: modelos, vistas y controladores. Este patrón permite lograr la separación de intereses. Con este patrón, las solicitudes del usuario se enrutan a un controlador que se encarga de trabajar con el modelo para realizar las acciones del usuario o recuperar los resultados de consultas. El controlador elige la vista para mostrar al usuario y proporciona cualquier dato de modelo que sea necesario (35).

En el siguiente diagrama se muestran los tres componentes principales y cuál hace referencia a los demás.



**Imagen 5.** Patrón Modelo-Vista-Controlador MVC. Fuente: Sommerville (2011).

### Modelo

El modelo en una aplicación de MVC representa el estado de la aplicación y cualquier lógica de negocios u operaciones que esta deba realizar. La lógica de negocios debe encapsularse en el modelo, junto con cualquier lógica de implementación para conservar el estado de la aplicación (35).

### Vistas

Las vistas se encargan de presentar el contenido a través de la interfaz de usuario. Debería haber la mínima lógica entre las vistas y cualquier lógica en ellas debe estar relacionada con la presentación de contenido (35).

### Controlador

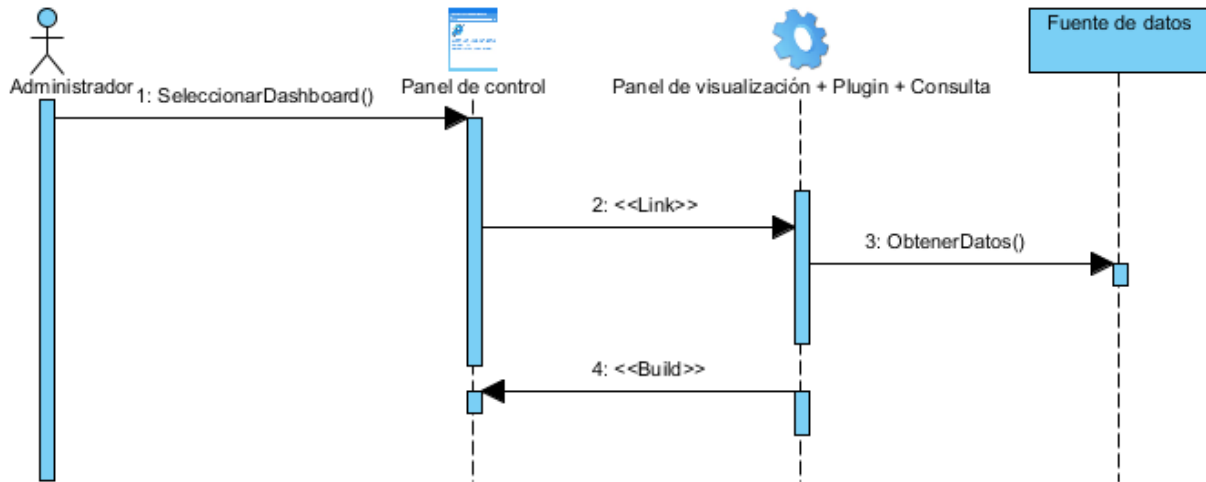
Los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y, en última instancia, seleccionan una vista para representarla. En una aplicación de MVC, la vista solo muestra información; el controlador controla y responde a la interacción y los datos que introducen los usuarios. En el patrón de MVC, el controlador es el punto de entrada inicial que se encarga de seleccionar con qué tipos de modelo trabajar y qué vistas representar (de ahí su nombre, ya que controla el modo en que la aplicación responde a una determinada solicitud) (35).

### **MVC en la propuesta de solución**

El sistema de visualización sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC).

- **Modelo (*Model*):** El modelo representa principalmente la capa de acceso a datos y la gestión de los mismos. Se integra con diferentes fuentes de datos, como bases de datos, servicios web o servicios de métricas en tiempo real. Estas fuentes de datos actúan como el modelo, ya que proporcionan los datos que se utilizan en los paneles y las visualizaciones. El sistema permite configurar y consultar estas fuentes de datos para obtener los datos necesarios para la visualización.
- **Vista (*View*):** La vista es lo que los usuarios ven y con lo que interactúan. Se ofrece una interfaz gráfica de usuario (GUI) intuitiva y flexible para crear y personalizar paneles y visualizaciones. Los paneles están compuestos de varios componentes visuales, como gráficos, tablas y paneles de estado, que se combinan para formar la vista final. Los usuarios pueden personalizar la apariencia y el diseño de las visualizaciones.
- **Controlador (*Controller*):** El controlador se encarga de gestionar las interacciones del usuario y de coordinar el flujo de datos entre el modelo y la vista. Los componentes como los paneles de visualización, *plugins* y las consultas de datos, interactúan entre sí y con el modelo y la vista para obtener y mostrar los datos requeridos. Aunque no hay una separación clara del controlador, los diferentes componentes internos funcionan juntos para llevar a cabo las operaciones necesarias para mostrar los datos en las visualizaciones.

A continuación, se muestra un diagrama de secuencia que muestra el comportamiento de este patrón.



**Imagen 6.** Diagrama de secuencia MVC del sistema de la propuesta de solución.  
Elaboración propia.

### 2.1.3. Principales módulos

Los módulos del sistema son fundamentales para habilitar su funcionalidad y flexibilidad. A continuación, se detallan los módulos principales que lo componen.

- **Fuentes de información:** las fuentes de información se refieren a los materiales, fuentes o recursos de los que se obtienen datos. Constituyen los referentes desde donde se toma la información procesada y visualizada en el sistema.
- **Servidor Grafana:** este módulo es el núcleo del sistema y se encarga de gestionar y servir la interfaz de usuario, las consultas a bases de datos, la autenticación de usuarios, y la configuración general del sistema. El servidor Grafana es el componente central que permite la interacción con la aplicación a través de un navegador web.
- **Interfaz de usuario:** la interfaz de usuario es el componente visible y accesible del sistema que permite a los usuarios interactuar con este. Incluye paneles de visualización de datos, tableros personalizables, herramientas de consulta y una variedad de opciones para la configuración y personalización de las visualizaciones.
- **Conectores de datos:** es compatible con una amplia gama de conectores de datos, que facilitan la integración con diversas fuentes de datos, como bases de datos de series temporales (por ejemplo, InfluxDB, Prometheus), bases de datos relacionales (por ejemplo, MySQL, PostgreSQL), sistemas de monitorización (por ejemplo, Zabbix), y



muchas otras fuentes de datos.

- Paneles y visualizaciones: estos módulos permiten la creación y visualización de diferentes tipos de datos a través de paneles personalizables. El sistema ofrece una amplia gama de visualizaciones, incluyendo gráficos de líneas, gráficos de barras, indicadores, mapas térmicos, tablas, y otros formatos que pueden ser configurados y combinados en un tablero (*dashboard*) para representar de forma intuitiva y efectiva la información.

Estos módulos constituyen la base del sistema y proporcionan una amplia gama de capacidades para la visualización y análisis de datos, así como la creación de paneles de control personalizados para el monitoreo.

#### **2.1.4. Flujo de información**

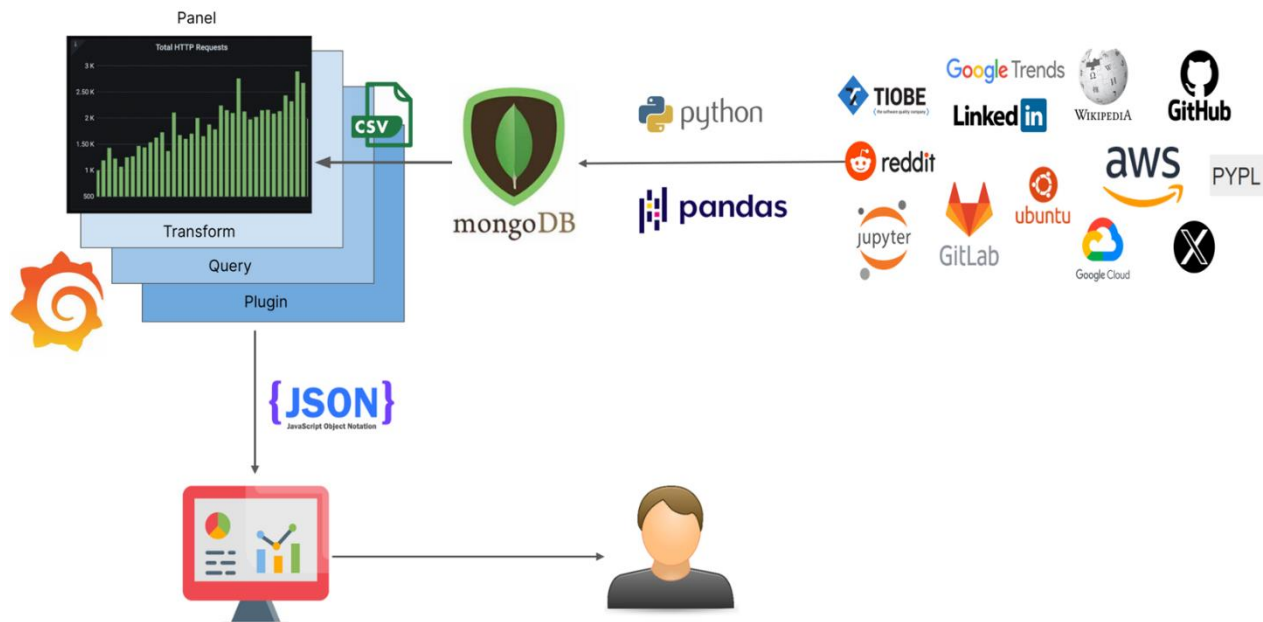
En este subepígrafe se explica cómo interactúan estos módulos y producen el flujo de información del sistema.

- Recopilación de datos: En primer lugar, se recopilan datos de diversas fuentes de información. Estas fuentes fueron identificadas en el capítulo anterior.
- Procesamiento de datos: En segundo lugar, se procesan los datos extraídos aplicando métodos y técnicas de análisis de datos. En este paso se emplea el lenguaje Python y su librería Pandas para el procesamiento de la información.
- Almacenamiento de datos: Luego de procesados, se definen las fuentes de datos para el almacenamiento de la información. Se emplea MongoDB y ficheros CSV.
- Conexión a fuentes de datos: Una vez que los datos se almacenan, se conecta a estas fuentes para extraer información. Utiliza conectores y *plugins* específicos para cada tipo de fuente de datos, lo que le permite obtener los datos necesarios para su visualización.
- Creación de paneles y gráficos: Luego, los usuarios pueden utilizar la interfaz para crear paneles y gráficos personalizados. Pueden seleccionar los datos de interés y elegir cómo quieren visualizarlos, ya sea a través de gráficos de líneas, barras, dispersión, mapas, tablas u otros tipos de visualizaciones.
- Distribución de los paneles: Finalmente, los paneles y gráficos creados se distribuyen a los usuarios o equipos relevantes. Estos paneles pueden ser incrustados en otros

sistemas o simplemente compartidos a través de enlaces, lo que permite a los usuarios acceder a la información de forma sencilla y clara.

A modo de conclusión, el flujo de información en el sistema implica la recopilación, procesamiento y almacenamiento de los datos, su conexión a fuentes específicas, la creación de visualizaciones personalizadas y la distribución de estos paneles para su consumo.

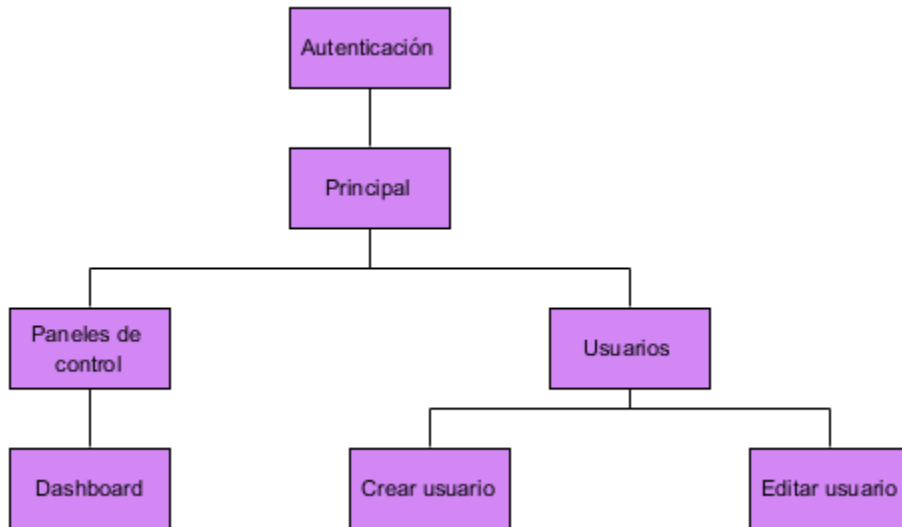
A continuación, se representa una imagen que describe el flujo de información en el sistema y su comportamiento.



**Imagen 7.** Flujo de información de la propuesta de solución. Elaboración propia.

### 2.1.5. Arquitectura de navegación

Se presenta el mapa de navegación del sistema. Este representa la estructura de las páginas, la arquitectura de la información y las relaciones entre diferentes elementos interactivos en la aplicación.



**Imagen 8.** Mapa de navegación del sistema. Elaboración propia.

El mapa representa cada una de las páginas que conforman la aplicación web y sus interrelaciones. Como primera pantalla se encuentra una página de autenticación. Una vez autenticados en el sistema, este nos redirige a una página principal. A partir de esta página principal, se puede redireccionar a otras.

Las más importantes de estas páginas serán descritas en el próximo capítulo, donde se abarcarán sus características, componentes y funcionamiento.

## **2.2. Requisitos del sistema**

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. (21) En la investigación estos se dividen en tres categorías: informacionales, funcionales, no funcionales.

### **2.2.1. Requisitos informacionales**

Los requisitos informacionales se refieren a las necesidades de información que deben ser satisfechas para la toma de decisiones efectivas y el desarrollo de procesos organizacionales (36). A continuación, se presentan los requisitos informacionales que debe cumplir el sistema de

visualización.

**RI 1.** Se requiere información que detalle la evolución de las tecnologías de desarrollo de software, como lenguajes de programación y bases de datos.

**RI 2.** La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.

**RI 3.** Garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.

**RI 4.** La información debe comprender diferentes formatos, como cadena de caracteres, enteros, series de tiempo, fechas.

**RI 5.** La información debe ser proveniente de fuentes confiables.

**RI 6.** La información debe proporcionar capacidades de análisis de datos, permitiendo a los usuarios extraer información y conocimientos significativos a partir de los datos almacenados.

**RI 7.** Los datos deben ser actualizados con una frecuencia mensual.

**RI 8.** La información debe ser presentada en formato gráfico y tabla.

**RI 9.** Se debe tener la capacidad de generar informes sobre los datos.

**RI 10.** Indicadores

Los requisitos informacionales RI en los *dashboards* están asociados a los indicadores que se seleccionen para ser analizados y visualizados, y el sistema debe permitir mostrar su comportamiento en el tiempo (11). Estos indicadores se seleccionan ya que aportan valor que permite medir la calidad de la tecnología en cuestión. Por lo tanto, se presentan los indicadores asociados a continuación.

- Nombre: nombre de la tecnología a analizar.
- Tipo: categoría subjetiva de tecnología donde encaja esta entidad. Por ejemplo: aplicación, lenguaje de programación, librería de programación, *framework* de programación, editor de código, sistema operativo.
- Cantidad de libros: recuento de libros sobre la tecnología en cuestión.
- Cantidad de artículos: recuento de artículos sobre la tecnología en cuestión.
- Cantidad de usuarios: estimación del número de usuarios que usa esta tecnología.
- Cantidad de proyectos: estimación del número de proyectos que existen sobre esta

tecnología.

- Cantidad de repositorios: cantidad de repositorios que reporta Github sobre esta tecnología.
- Total: sumatoria de todos los indicadores anteriores. Indispensable para establecer el ranking de popularidad.
- Conteo de búsquedas de tutoriales en Google (tomado de PYPL): conteo de búsquedas de tutoriales sobre el lenguaje de programación en Google.
- Conteo de búsquedas de la página de descarga en Google (tomado de PYPL): conteo de búsquedas de las páginas de descarga sobre el IDE en Google.
- Conteo de búsquedas del nombre en Google (tomado de PYPL): conteo de búsquedas del nombre de la base de datos en Google.
- Conteo de búsquedas del nombre en múltiples motores de búsqueda (tomado de TIOBE): conteo de búsquedas del nombre del lenguaje de programación en varios motores de búsqueda.

### **Calificación del ranking**

Para el establecimiento de uno de los tres sistemas de ranking se toma en cuenta el indicador Total, que hace referencia a la sumatoria de todos los indicadores tecnológicos que le anteceden en la lista. Posteriormente, se realiza el orden de los elementos de manera decreciente, de forma tal que queden más arriba los elementos con mayor valor. Para la definición de un segundo y tercer ranking, se usan los datos e indicadores brindados por TIOBE y PYPL.

### **2.2.2. Requisitos funcionales**

Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema (21). Mediante el empleo de la técnica de generación de ideas *brainstorming*, los métodos analítico-sintético, análisis documental y las interacciones con el cliente, se definen los requisitos funcionales que debe cumplir el sistema de visualización.

**Tabla 4.** Requisitos funcionales

No.	Nombre	Descripción	Prioridad	Complejidad
<b>RF 1</b>	Autenticar	El sistema debe permitir al usuario autenticarse, usando nombre de usuario y contraseña.	Alta	Baja
<b>RF 2</b>	Cerrar sesión	El sistema debe permitir al usuario cerrar su sesión.	Alta	Baja
<b>RF 3</b>	Crear panel de control	El sistema debe permitir crear los paneles de control requeridos.	Alta	Baja
<b>RF 4</b>	Visualizar panel de control	El sistema debe permitir visualizar los paneles de control.	Alta	Baja
<b>RF 5</b>	Eliminar panel de control	El sistema debe permitir eliminar los paneles de control creados.	Alta	Baja
<b>RF 6</b>	Importar panel de control	El sistema debe permitir importar un panel de control previamente creado en formato JSON.	Baja	Baja
<b>RF 7</b>	Exportar panel de control	El sistema debe permitir exportar un panel de control en formato JSON.	Alta	Baja
<b>RF 8</b>	Listar paneles de control	El sistema debe permitir listar los paneles de control creados.	Alta	Baja
<b>RF 9</b>	Buscar panel de control	El sistema debe permitir buscar los paneles de control creados.	Baja	Baja
<b>RF 10</b>	Añadir visualización	El sistema debe permitir agregar una visualización	Alta	Baja

		gráfica al panel de control seleccionado.		
<b>RF 11</b>	Manipular visualización	El sistema debe ser capaz de permitirle al usuario interactuar y manipular la visualización gráfica.	Baja	Baja
<b>RF 12</b>	Seleccionar fuente de datos	El sistema debe permitir, desde la visualización gráfica, seleccionar la fuente de datos que será usada para mostrar la información en el gráfico seleccionado.	Alta	Baja
<b>RF 13</b>	Configurar visualización	El sistema debe ser capaz de permitirle al usuario configurar la visualización gráfica.	Alta	Alta
<b>RF 14</b>	Eliminar visualización	El sistema debe permitir eliminar la visualización gráfica.	Alta	Baja
<b>RF 15</b>	Crear usuario	El sistema debe dar la posibilidad de crear usuarios.	Alta	Baja
<b>RF 16</b>	Modificar usuario	El sistema debe dar la posibilidad de modificar los parámetros de usuario.	Alta	Baja
<b>RF 17</b>	Eliminar usuario	El sistema debe dar la posibilidad de eliminar usuarios.	Alta	Baja
<b>RF 18</b>	Asignar rol a usuario	El sistema debe permitir asignarle un rol al usuario según sus privilegios en el sistema.	Alta	Baja
<b>RF 19</b>	Listar usuarios	El sistema debe permitir listar los usuarios registrados.	Alta	Baja

<b>RF 20</b>	Buscar usuario	El sistema debe permitir buscar los usuarios del sistema.	Alta	Baja
<b>RF 21</b>	Añadir fuente de datos	El sistema debe dar la posibilidad de añadir una fuente de datos para las visualizaciones.	Alta	Baja
<b>RF 22</b>	Eliminar fuente de datos	El sistema debe permitir eliminar la fuente de datos.	Alta	Baja
<b>RF 23</b>	Listar fuente de datos	El sistema debe permitir listar las fuentes de datos creadas.	Alta	Baja
<b>RF 24</b>	Buscar fuente de datos	El sistema debe permitir buscar las fuentes de datos disponibles.	Alta	Baja
<b>RF 25</b>	Exportar datos	El sistema debe permitir exportar la información de un gráfico en formato CSV.	Alta	Baja
<b>RF 26</b>	Generar imagen de gráfico	El sistema debe ser capaz de renderizar una imagen del gráfico seleccionado y permitir su descarga.	Alta	Baja

### 2.2.3. Requisitos no funcionales

Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (21). A continuación, se describen los requisitos no funcionales que debe cumplir el sistema.

#### Requisito(s) de Usabilidad

**RnF 1.** Tipo de aplicación informática: aplicación web.

**RnF 2.** Finalidad: monitorear tendencias de investigación mediante el uso de indicadores que posibiliten el análisis y visualización de datos. La aplicación debe proporcionar al usuario



información relevante sobre temas de investigación. El sistema debe apoyar la toma de decisiones mediante un conjunto de funcionalidades que se ajusten a las necesidades de los usuarios.

### **Requisito(s) de Rendimiento**

**RnF 3.** Debe ser capaz de alojar a todos los ordenadores disponibles en el centro, de forma simultánea.

### **Requisito(s) de Software**

**RnF 4.** Poseer algún navegador web instalado en la estación de trabajo del usuario. Se recomienda el uso de Chrome, Firefox, Safari, Edge. Es posible que las versiones anteriores al año 2022 de estos navegadores no sean compatibles, por lo que siempre se debe actualizar a la última versión del navegador.

**RnF 5.** El navegador debe tener habilitado JavaScript.

**RnF 6.** Compatible con los siguientes sistemas operativos: Debian o Ubuntu, Red Hat, RHEL o Fedora, SUSE u openSUSE, macOS, Windows.

### **Requisito(s) de Hardware**

**RnF 7.** El ordenador debe contar con al menos 2Gb de memoria RAM.

**RnF 8.** El ordenador debe tener un mínimo de 80Gb de almacenamiento.

**RnF 9.** Debe contar como mínimo con un procesador Intel Pentium Core2Duo.

### **Requisito(s) de Seguridad**

**RnF 10.** Protección contra acciones no autorizadas o que puedan afectar la integridad del sistema. Crear cuentas de usuarios que contengan roles y permisos sobre el sistema.

**RnF 11.** El sitio debe emplear protocolo https.

**RnF 12.** Los usuarios del sistema deben identificarse con UUID.

### **Requisito(s) de Restricciones del diseño y la implementación**

**RnF 13.** El sistema será implementado en Grafana como herramienta de visualización.

**RnF 14.** El sistema usará MongoDB como gestor de base de datos.

**RnF 15.** Se empleará el lenguaje de programación Python.

**RnF 16.** Se usará Visual Studio Code como entorno de desarrollo integrado.

**RnF 17.** El sistema será modelado en la herramienta Visual Paradigm para UML para los artefactos ingenieriles.

### 2.3. Historias de usuario

Las historias de usuario describen la salida necesaria, características y funcionalidad del software que se va a elaborar. Una historia de usuario es una explicación informal de una función de software, escrita desde la perspectiva del usuario final (37, 38). La metodología XP seleccionada para la realización de este proyecto emplea historias de usuario como uno de sus artefactos ingenieriles. A continuación, se representan las descripciones de las historias de usuario sobre los requisitos funcionales más importantes.

**Tabla 5.** Historia de usuario Autenticar

<b>No:</b> 1	<b>Nombre:</b> Autenticar
<b>Usuario:</b> analista	
<b>Prioridad:</b> alta	<b>Complejidad:</b> baja
<b>Descripción:</b> El sistema debe permitir al usuario autenticarse, usando nombre de usuario y contraseña.	
<b>Observaciones:</b> Al insertar los datos incorrectamente, el sistema debe lanzar un mensaje advirtiendo sobre el error.	

**Tabla 6.** Historia de usuario Cerrar sesión

<b>No:</b> 2	<b>Nombre:</b> Cerrar sesión
<b>Usuario:</b> analista	
<b>Prioridad:</b> alta	<b>Complejidad:</b> baja
<b>Descripción:</b> El sistema debe permitir al usuario cerrar su sesión.	
<b>Observaciones:</b>	

**Tabla 7.** Historia de usuario Visualizar panel de control

<b>No:</b> 3	<b>Nombre:</b> Visualizar panel de control
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe permitir visualizar los paneles de control.	
<b>Observaciones:</b> El panel de control debe estar previamente creado.	

**Tabla 8.** Historia de usuario Listar paneles de control

<b>No:</b> 4	<b>Nombre:</b> Listar paneles de control
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe permitir listar los paneles de control creados.	
<b>Observaciones:</b> Los paneles deben estar previamente creados para ser listados.	

**Tabla 9.** Historia de usuario Manipular visualización

<b>No:</b> 5	<b>Nombre:</b> Manipular visualización
<b>Usuario:</b> analista	
<b>Prioridad:</b> Baja	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe ser capaz de permitirle al usuario interactuar y manipular la visualización gráfica.	
<b>Observaciones:</b> La visualización debe estar previamente creada.	

**Tabla 10.** Historia de usuario Seleccionar fuente de datos

<b>No:</b> 6	<b>Nombre:</b> Seleccionar fuente de datos
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja

<b>Descripción:</b> El sistema debe permitir, desde la visualización gráfica, seleccionar la fuente de datos que será usada para mostrar la información en el gráfico seleccionado.
<b>Observaciones:</b> La fuente de datos debe estar creada previamente.

**Tabla 11.** Historia de usuario Crear usuario

<b>No:</b> 7	<b>Nombre:</b> Crear usuario
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe dar la posibilidad de crear usuarios.	
<b>Observaciones:</b>	

**Tabla 12.** Historia de usuario Modificar usuario

<b>No:</b> 8	<b>Nombre:</b> Modificar usuario
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe dar la posibilidad de modificar los parámetros de usuario.	
<b>Observaciones:</b> El usuario debe estar previamente creado.	

**Tabla 13.** Historia de usuario Eliminar usuario

<b>No:</b> 9	<b>Nombre:</b> Eliminar usuario
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe dar la posibilidad de eliminar el usuario.	
<b>Observaciones:</b> El usuario debe estar previamente creado.	

**Tabla 14.** Historia de usuario Asignar rol a usuario

<b>No:</b> 10	<b>Nombre:</b> Asignar rol a usuario
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe permitir asignarle un rol al usuario según sus privilegios en el sistema.	
<b>Observaciones:</b> El usuario debe estar previamente creado.	

**Tabla 15.** Historia de usuario Listar usuario

<b>No:</b> 11	<b>Nombre:</b> Listar usuario
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe permitir listar los usuarios registrados.	
<b>Observaciones:</b> Debe existir al menos un usuario creado para poder listar.	

**Tabla 16.** Historia de usuario Exportar datos

<b>No:</b> 12	<b>Nombre:</b> Exportar datos
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe permitir exportar la información de un gráfico en formato CSV.	
<b>Observaciones:</b> La visualización gráfica debe estar creada y tener asignada una fuente de datos con información.	

**Tabla 17.** Historia de usuario Generar imagen de gráfico

<b>No:</b> 12	<b>Nombre:</b> Generar imagen de gráfico
<b>Usuario:</b> analista	
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Baja
<b>Descripción:</b> El sistema debe ser capaz de renderizar una imagen del gráfico seleccionado y permitir su descarga.	
<b>Observaciones:</b> La visualización gráfica debe estar creada previamente.	

### Conclusiones parciales

A lo largo de este capítulo se describe el proceso de análisis y diseño del sistema de visualización. Se identifican un total de 10 requisitos informacionales, 26 funcionales y 17 no funcionales. A partir de la definición de las historias de usuario se pudieron identificar las principales funcionalidades que debe cumplir el sistema, por lo que se redactan las historias de usuario sobre los requisitos funcionales más importantes. La utilización de la herramienta Grafana facilita la selección del patrón MVC como patrón de diseño que refleja el comportamiento del sistema de visualización. Este comportamiento se describe mediante el uso de un diagrama de secuencia. Además, el uso de esta herramienta también determina la arquitectura cliente-servidor que utiliza el sistema. El mapa de navegación describe las páginas que conforman la aplicación web y determina el flujo de trabajo que desempeña el usuario al navegar sobre el sistema. El flujo de información describe cómo ocurre el funcionamiento de la plataforma.

# Capítulo III. Implementación y prueba del sistema de visualización de indicadores para la vigilancia tecnológica

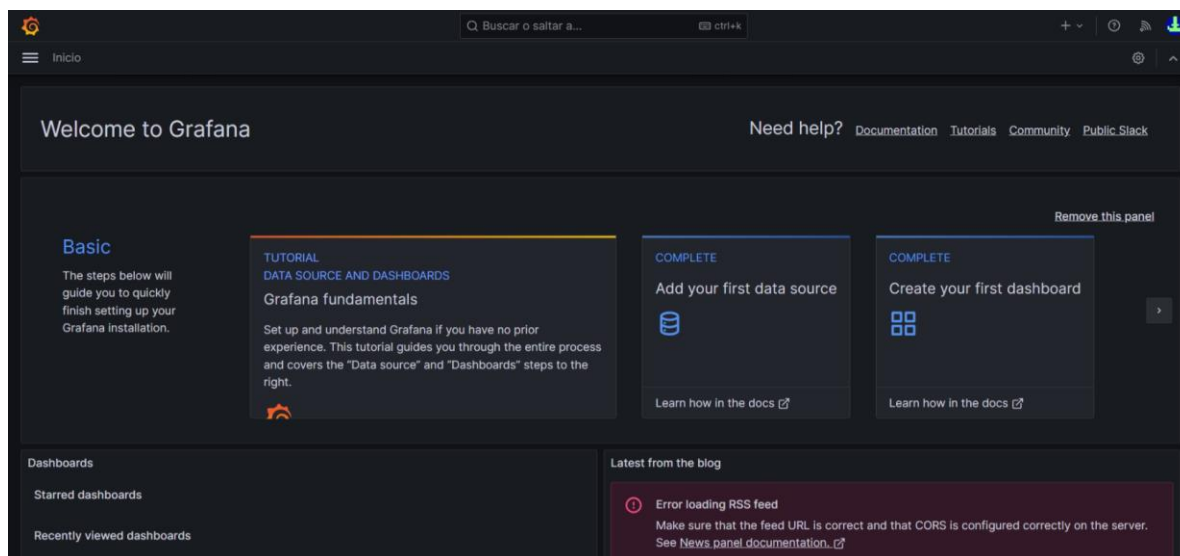
## Introducción al capítulo

La implementación de software es la fase en la que el software se desarrolla, se integra, se prueba y finalmente se instala y se pone en funcionamiento en un entorno de producción. Las pruebas de software son el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. El siguiente capítulo trata sobre el proceso de implementación de la propuesta de solución y realización de pruebas de software sobre el sistema. Se explica el funcionamiento del sistema, teniendo en cuenta la descripción de algunas de sus páginas más importantes y el diagrama de despliegue. Se define la estrategia de pruebas y el diseño de casos de prueba. Se muestran los resultados de estas últimas y se arriba a conclusiones.

## 3.1. Funcionamiento del sistema de visualización

### 3.1.1. Descripción de los contenidos de la página principal

En la cabecera, esquina superior derecha, se ubica el usuario autenticado en el instante en que se usa el sistema. Al hacerle clic, se despliega un menú de opciones, que incluye el botón Cerrar sesión. En el centro de la cabecera se ubica una barra de búsqueda. En la esquina superior izquierda se encuentra el logo de la herramienta de visualización, y bajo este, un botón que activa/desactiva la barra lateral con un conjunto de opciones. En esta barra se encuentran dos opciones importantes, Paneles de control y Usuarios. Paneles de control redirecciona a la pantalla donde se encuentran listados los *dashboards*, mientras que Usuarios lo hace hacia el área de trabajo de administración de usuarios. En el área de trabajo se encuentran accesos rápidos a la documentación y tutoriales sobre la herramienta de visualización, así como los *dashboards* vistos recientemente.

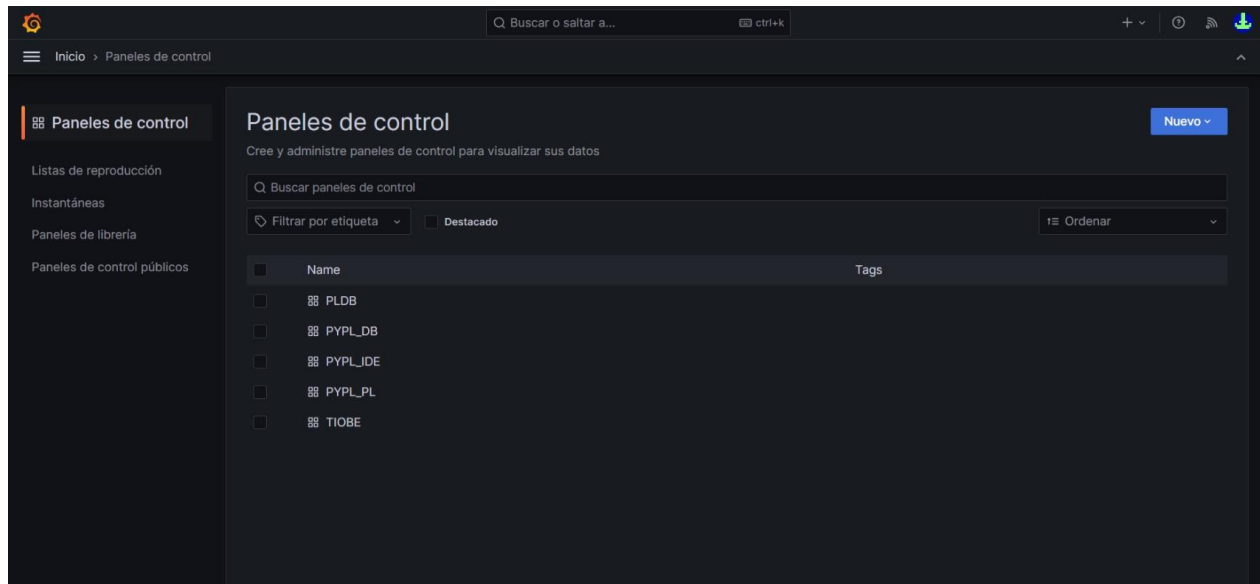


**Imagen 9.** Vista de la página principal. Captura de pantalla del sistema.

### 3.1.2. Descripción de los contenidos de la página Paneles de control

En la cabecera, esquina superior derecha, se ubica el usuario autenticado en el instante en que se usa el sistema. Al hacerle clic, se despliega un menú de opciones, que incluye el botón Cerrar sesión. En el centro de la cabecera se ubica una barra de búsqueda. En la esquina superior izquierda se encuentra el logo de la herramienta de visualización, y bajo este, un botón que activa/desactiva la barra lateral con un conjunto de opciones. En esta barra se encuentran dos opciones importantes, Paneles de control y Usuarios. Paneles de control redirecciona a la pantalla donde se encuentran listados los *dashboards*, mientras que Usuarios lo hace hacia el área de trabajo de administración de usuarios. En el área de trabajo se encuentran listados los *dashboards* creados y una barra de búsqueda para estos. En la esquina superior derecha de esta área se ubica el botón Nuevo, que permite crear un nuevo panel de control.





**Imagen 10.** Vista de la página Paneles de control. Captura de pantalla del sistema.

### 3.1.3. Descripción de los contenidos de la página Usuarios

En la cabecera, esquina superior derecha, se ubica el usuario autenticado en el instante en que se usa el sistema. Al hacerle clic, se despliega un menú de opciones, que incluye el botón Cerrar sesión. En el centro de la cabecera se ubica una barra de búsqueda. En la esquina superior izquierda se encuentra el logo de la herramienta de visualización, y bajo este, un botón que activa/desactiva la barra lateral con un conjunto de opciones. En esta barra se encuentran dos opciones importantes, Paneles de control y Usuarios. Paneles de control redirecciona a la pantalla donde se encuentran listados los *dashboards*, mientras que Usuarios lo hace hacia el área de trabajo de administración de usuarios. En el área de trabajo se encuentran listados los usuarios creados y una barra de búsqueda para estos. En la esquina superior derecha de esta área se ubica el botón Nuevo usuario, que permite crear un nuevo usuario.

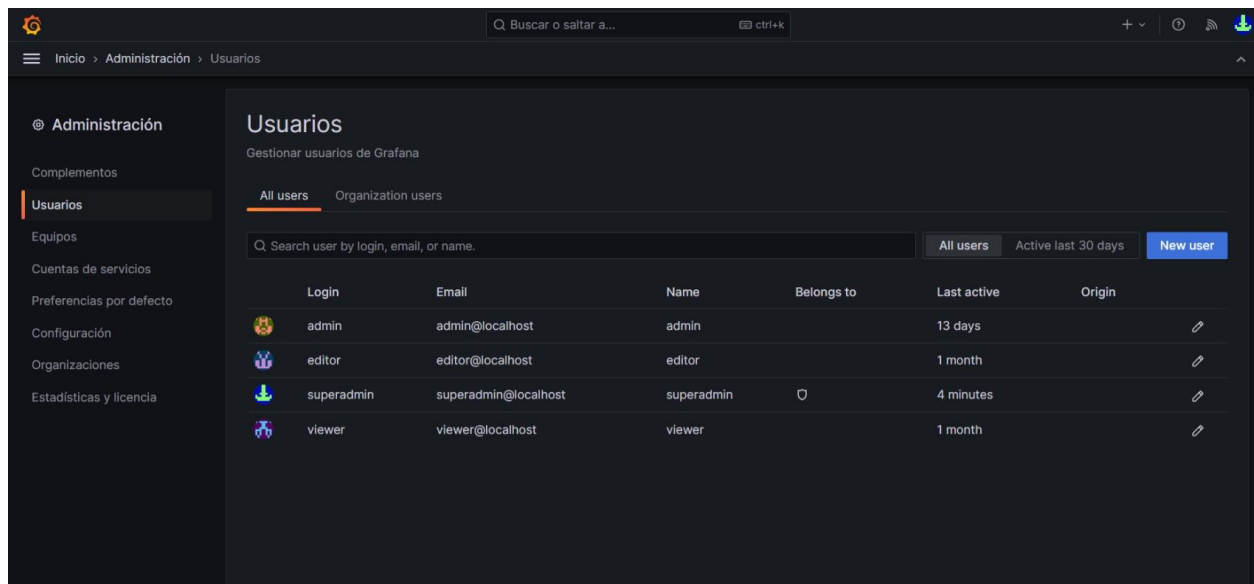
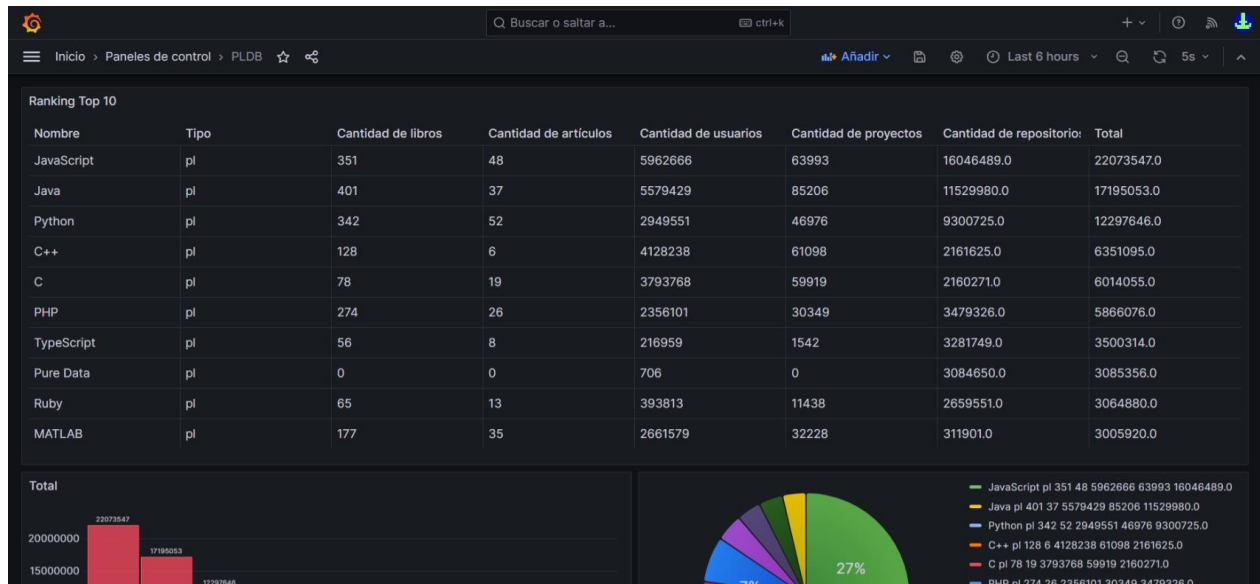


Imagen 11. Vista de la página Usuarios. Captura de pantalla del sistema.

### 3.1.4. Descripción de los contenidos de la página Dashboard

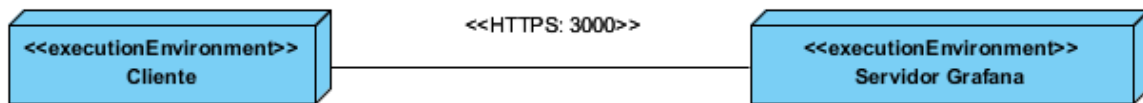
En la cabecera, esquina superior derecha, se ubica el usuario autenticado en el instante en que se usa el sistema. Al hacerle clic, se despliega un menú de opciones, que incluye el botón Cerrar sesión. En el centro de la cabecera se ubica una barra de búsqueda. En la esquina superior izquierda se encuentra el logo de la herramienta de visualización, y bajo este, un botón que activa/desactiva la barra lateral con un conjunto de opciones. En esta barra se encuentran dos opciones importantes, Paneles de control y Usuarios. Paneles de control redirecciona a la pantalla donde se encuentran listados los *dashboards*, mientras que Usuarios lo hace hacia el área de trabajo de administración de usuarios. En esta página, en la cabecera, se encuentra una barra de herramientas con múltiples opciones. En la esquina derecha se encuentra un botón Añadir, que permite agregar una nueva visualización al *dashboard*. Además, se encuentran las opciones Guardar y Configurar *dashboard*, que como sus nombres lo indican, guarda los cambios realizados y permite configurar el tablero en cuestión respectivamente. De último, tiene un botón que permite administrar la frecuencia de actualización de los datos en el panel. En la esquina izquierda se ubica el nombre del panel de control que está siendo visualizado y el botón Compartir. Este último permite compartir un enlace al *dashboard*, de manera pública o solo para los usuarios del sistema. Además, permite exportar el *dashboard* en formato JSON. El área de trabajo abarca todas las visualizaciones creadas.



**Imagen 12.** Vista de la página Dashboard. Captura de pantalla del sistema.

### 3.1.5. Diagrama de despliegue

El diagrama de despliegue muestra la relación entre los componentes de software y el hardware en el que se ejecutan. Muestra cómo se distribuyen físicamente los componentes del sistema, como servidores, computadoras, dispositivos móviles, redes y otros dispositivos de hardware. A continuación, se representa el diagrama de despliegue del sistema de visualización.



**Imagen 13.** Diagrama de despliegue del sistema de visualización. Elaboración propia.

En la figura representada anteriormente se evidencia la existencia de un nodo Cliente, que hace referencia al usuario que interactúa con el sistema. El nodo Servidor Grafana se refiere al servidor desplegado de la herramienta Grafana, la cual ayuda en la creación del sistema de visualización. El cliente es capaz de conectarse a este servidor mediante un navegador, empleando protocolo https por el puerto 3000.

### 3.2. Estrategia de pruebas

La estrategia de prueba describe el enfoque y los objetivos generales de las actividades de prueba. Incluye los niveles de prueba a aplicar, los tipos de prueba a ejecutar, herramientas empleadas y los casos de prueba diseñados para lograr los objetivos.

Define:

- Técnicas de pruebas (manual o automática) y herramientas a utilizar.
- Criterios de éxitos y culminación de las pruebas.
- Consideraciones especiales relacionadas con los recursos.

#### Metas

- Comprobar el funcionamiento de las operaciones que contiene la aplicación.
- Verificar el comportamiento de la eficiencia, seguridad y el rendimiento de la aplicación, así como el cumplimiento de los parámetros especificados para el sistema.

**Tabla 18.** Tipos y niveles de pruebas

Tipo/Nivel de prueba	Objetivo
Alfa/Aceptación	Evaluar si el sistema o software cumple con los requisitos establecidos por el cliente o los usuarios y si satisface sus necesidades y expectativas.
Rendimiento/Integridad	Evaluar y medir el desempeño y la capacidad de respuesta de un sistema, aplicación o infraestructura bajo diferentes condiciones de carga y estrés.
Seguridad/Integridad	Evaluar y verificar la capacidad del sistema para resistir y protegerse contra posibles amenazas y vulnerabilidades de seguridad.

**Tabla 19.** Estrategia de pruebas

Nivel de prueba	Tipo de prueba	Método	Herramientas	Roles	Objetivos de los niveles
Sistema	Rendimiento	Caja negra	Grafana K6 Postman	Tester	Validar que el sistema está completo y funciona como se espera. Verificar que los requisitos funcionales y no funcionales del sistema son los especificados.
	Seguridad		OWASP ZAP Postman		
	Portabilidad		Navegadores web Sistemas operativos		
Aceptación	Alfa	Caja Negra	Casos de prueba	Analistas y usuarios finales	

### 3.3. Diseño de casos de prueba por tipo y resultados

#### 3.3.1. Pruebas alfa

**Tabla 20.** Prueba de aceptación #1

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Autenticar	
<b>Descripción:</b> Comprobar que los datos de inicio de sesión introducidos son correctos.	

<b>Condiciones de ejecución:</b> El usuario debe introducir sus credenciales: nombre de usuario y contraseña.
<b>Pasos de ejecución:</b> Escribir nombre de usuario y contraseña en el formulario requerido y enviar la respuesta.
<b>Resultado:</b> Satisfactorio.

**Tabla 21.** Prueba de aceptación #2

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Autenticar.	
<b>Descripción:</b> Comprobar que los datos de inicio de sesión introducidos son incorrectos.	
<b>Condiciones de ejecución:</b> El usuario debe introducir sus credenciales: nombre de usuario y contraseña.	
<b>Pasos de ejecución:</b> Escribir nombre de usuario y contraseña en el formulario requerido y enviar la respuesta.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 22.** Prueba de aceptación #3

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Cerrar sesión.	
<b>Descripción:</b> Comprobar que se cierra la sesión correctamente.	
<b>Condiciones de ejecución:</b> El usuario debe encontrarse autenticado en el sistema.	
<b>Pasos de ejecución:</b> El usuario debe seleccionar el botón Cerrar sesión.	

<b>Resultado:</b> Satisfactorio.
----------------------------------

**Tabla 23.** Prueba de aceptación #4

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3_P1	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Visualizar panel de control.	
<b>Descripción:</b> Comprobar que el usuario puede visualizar los paneles requeridos.	
<b>Condiciones de ejecución:</b> El panel debe estar previamente creado.	
<b>Pasos de ejecución:</b> El usuario debe seleccionar el panel requerido en la lista de paneles.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 24.** Prueba de aceptación #5

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Listar paneles de control.	
<b>Descripción:</b> Comprobar que el sistema lista los paneles.	
<b>Condiciones de ejecución:</b> Debe existir al menos un panel creado.	
<b>Pasos de ejecución:</b> El usuario debe ir a la opción Paneles de control ubicada en la barra lateral del sitio.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 25.** Prueba de aceptación #6

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P1	<b>Historia de Usuario:</b> 5

<b>Nombre:</b> Manipular visualización.
<b>Descripción:</b> Comprobar que el usuario puede manipular e interactuar con la visualización gráfica del panel.
<b>Condiciones de ejecución:</b> La visualización debe estar previamente creada.
<b>Pasos de ejecución:</b> El usuario debe arrastrar el panel a la posición que desee, cambiar su tamaño, e interactuar con las opciones que brinda.
<b>Resultado:</b> Satisfactorio.

**Tabla 26.** Prueba de aceptación #7

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Seleccionar fuente de datos.	
<b>Descripción:</b> Comprobar que el usuario puede seleccionar la fuente de datos que será usada en la visualización.	
<b>Condiciones de ejecución:</b> Debe existir, al menos, una fuente de datos creada y una visualización.	
<b>Pasos de ejecución:</b> Seleccionar la visualización requerida y dar en el botón Editar. Luego seleccionar la fuente de datos en la consulta ubicada debajo de la visualización.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 27.** Prueba de aceptación #8

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU7_P1	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Crear usuario.	
<b>Descripción:</b> Comprobar que se pueden crear usuarios en el sistema.	



<b>Condiciones de ejecución:</b> Poseer los permisos de administrador en el sistema.
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar la opción Nuevo usuario e insertar los datos correspondientes.
<b>Resultado:</b> Satisfactorio.

**Tabla 28.** Prueba de aceptación #9

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU7_P2	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Crear usuario.	
<b>Descripción:</b> Comprobar que no se puede crear un usuario ya existente.	
<b>Condiciones de ejecución:</b> Poseer los permisos de administrador en el sistema.	
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar la opción Nuevo usuario e insertar los datos correspondientes.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 29.** Prueba de aceptación #10

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU8_P1	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Modificar usuario.	
<b>Descripción:</b> Comprobar que se pueden modificar los parámetros de usuario.	
<b>Condiciones de ejecución:</b> Poseer los permisos de administrador del sistema. Usuario existente.	
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar sobre el usuario ya creado y modificar sus parámetros.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 30.** Prueba de aceptación #11

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU8_P2	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Modificar usuario.	
<b>Descripción:</b> Comprobar que no se puede modificar un usuario por uno ya existente.	
<b>Condiciones de ejecución:</b> Poseer los permisos de administrador del sistema. Usuario existente.	
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar sobre el usuario ya creado y modificar sus parámetros.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 31.** Prueba de aceptación #12

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU9_P1	<b>Historia de Usuario:</b> 9
<b>Nombre:</b> Eliminar usuario.	
<b>Descripción:</b> Comprobar que se puede eliminar un usuario creado.	
<b>Condiciones de ejecución:</b> Poseer los permisos de administrador del sistema. Usuario existente.	
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar sobre el usuario ya creado y eliminarlo.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 32.** Prueba de aceptación #13

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU10_P1	<b>Historia de Usuario:</b> 10

<b>Nombre:</b> Asignar rol al usuario.
<b>Descripción:</b> Comprobar que se le puede asignar un rol al usuario.
<b>Condiciones de ejecución:</b> Poseer permisos de administrador del sistema. Usuario creado.
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio. Seleccionar sobre el usuario ya creado y asignarle un rol.
<b>Resultado:</b> Satisfactorio.

**Tabla 33.** Prueba de aceptación #14

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU11_P1	<b>Historia de Usuario:</b> 11
<b>Nombre:</b> Listar usuarios.	
<b>Descripción:</b> Comprobar que el sistema lista los usuarios creados.	
<b>Condiciones de ejecución:</b> Poseer permisos de administrador del sistema. Existir al menos un usuario creado.	
<b>Pasos de ejecución:</b> Acceder al menú Usuarios ubicado en la barra lateral del sitio.	
<b>Resultado:</b> Satisfactorio.	

**Tabla 34.** Prueba de aceptación #15

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU12_P1	<b>Historia de Usuario:</b> 12
<b>Nombre:</b> Generar imagen de gráfico.	
<b>Descripción:</b> Comprobar que el sistema permite generar una imagen del gráfico seleccionado.	
<b>Condiciones de ejecución:</b> Existir una visualización.	

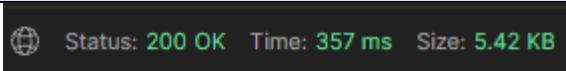
<b>Pasos de ejecución:</b> Seleccionar el gráfico requerido y marcar la opción Compartir. Posteriormente, generar un enlace con imagen renderizada.
<b>Resultado:</b> Satisfactorio.

**Tabla 35.** Prueba de aceptación #16

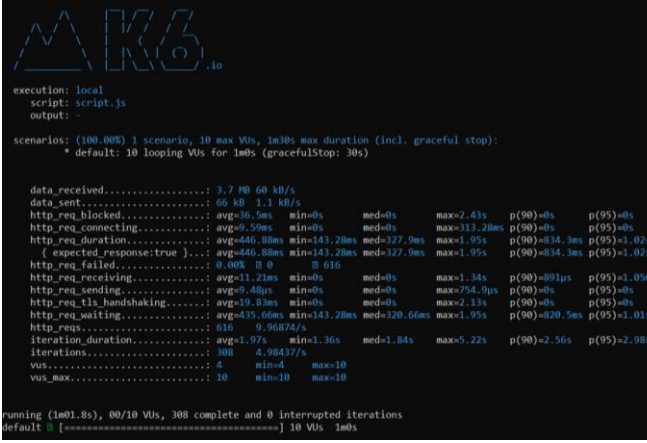
<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU13_P1	<b>Historia de Usuario:</b> 13
<b>Nombre:</b> Exportar datos.	
<b>Descripción:</b> Comprobar que el sistema permite descargar la información de un gráfico.	
<b>Condiciones de ejecución:</b> Existir una visualización. Debe estar vinculada a una fuente de datos.	
<b>Pasos de ejecución:</b> Seleccionar el gráfico requerido y marcar la opción Inspeccionar. Posteriormente, señalar la opción Descargar CSV.	
<b>Resultado:</b> Satisfactorio.	

### 3.3.2. Pruebas de rendimiento

**Tabla 36.** Pruebas de rendimiento

ID del Escenario	Escenario	Herramienta	Respuesta de la herramienta	Resultado
ER 1	Tiempo de respuesta del sistema a una petición	Postman		El sistema responde a la solicitud con un tiempo de 357ms.

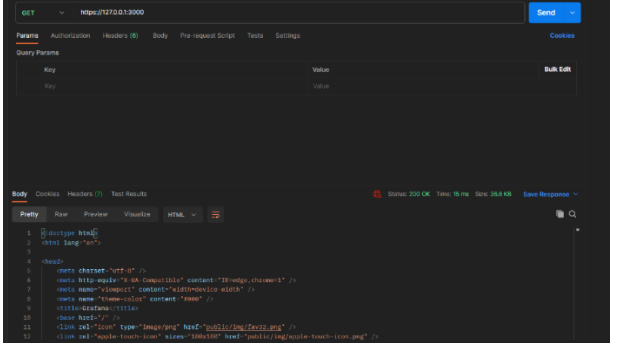
	de acceso.			Este valor indica el tiempo de respuesta del sistema a una petición de acceso por parte de un usuario.																				
ER 2	Consumo en navegadores.	Navegadores	<table border="1"> <thead> <tr> <th>Process</th> <th>CPU</th> <th>Private Bytes</th> <th>Disk I/O</th> <th>Network I/O</th> </tr> </thead> <tbody> <tr> <td>Google Chrome (8)</td> <td>0%</td> <td>278.5 MB</td> <td>0.5 MB/s</td> <td>0 Mbps</td> </tr> <tr> <td>Firefox (10)</td> <td>0%</td> <td>604.6 MB</td> <td>5.9 MB/s</td> <td>0 Mbps</td> </tr> <tr> <td>Grafana y 1 página más - Perfil...</td> <td>0.8%</td> <td>213.3 MB</td> <td>0.1 MB/s</td> <td>0 Mbps</td> </tr> </tbody> </table>	Process	CPU	Private Bytes	Disk I/O	Network I/O	Google Chrome (8)	0%	278.5 MB	0.5 MB/s	0 Mbps	Firefox (10)	0%	604.6 MB	5.9 MB/s	0 Mbps	Grafana y 1 página más - Perfil...	0.8%	213.3 MB	0.1 MB/s	0 Mbps	Consumo en Chrome, Edge y Firefox. Se muestran los valores de uso de CPU, memoria RAM, disco y red respectivamente.
Process	CPU	Private Bytes	Disk I/O	Network I/O																				
Google Chrome (8)	0%	278.5 MB	0.5 MB/s	0 Mbps																				
Firefox (10)	0%	604.6 MB	5.9 MB/s	0 Mbps																				
Grafana y 1 página más - Perfil...	0.8%	213.3 MB	0.1 MB/s	0 Mbps																				

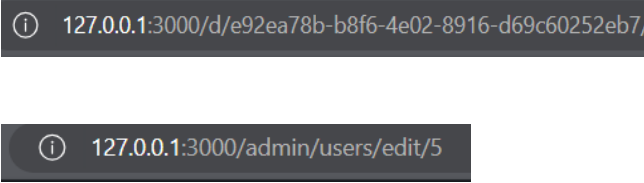
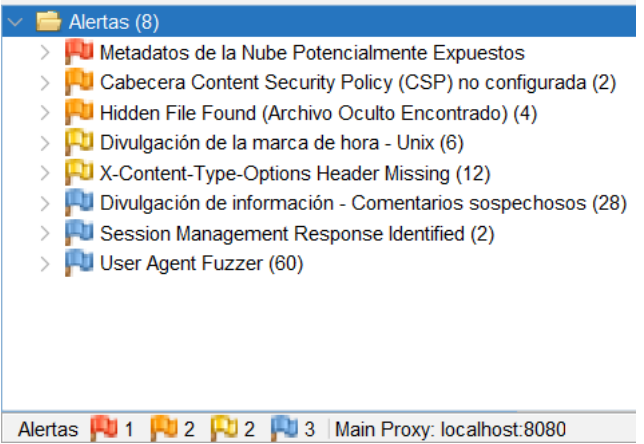
ER 3	Tiempo de respuesta del sistema a peticiones de 10 usuarios simultáneamente en un tiempo de 60 segundos.	Grafana K6	 <pre> execution: local script: script.js output: -  scenarios: (100.00%) 1 scenario, 10 max VUs, 1m30s max duration (incl. graceful stop):  * default: 10 looping VUs for 1m0s (gracefulStop: 30s)  data_received.....: 3.7 MB 60 kB/s data_sent.....: 66 kB 1.1 kB/s http_req_blocked.....: avg=16.5ms min=0s med=0s max=2.43s p(90)=0s p(95)=0s http_req_connecting.....: avg=9.59ms min=0s med=0s max=113.28ms p(90)=0s p(95)=0s http_req_duration.....: avg=446.88ms min=143.28ms med=327.9ms max=1.95s p(90)=834.3ms p(95)=1.02s  { expected_response:true }...: avg=446.88ms min=143.28ms med=327.9ms max=1.95s p(90)=834.3ms p(95)=1.02s http_req_failed.....: 0.00% 0 0 0 616 http_req_receiving.....: avg=11.21ms min=0s med=0s max=1.34s p(90)=891µs p(95)=1.05ms http_req_sending.....: avg=7.46µs min=0s med=0s max=794.9µs p(90)=0s p(95)=0s http_req_tls_handshaking.....: avg=19.03ms min=0s med=0s max=2.13s p(90)=0s p(95)=0s http_req_waiting.....: avg=435.66ms min=143.28ms med=320.66ms max=1.95s p(90)=820.5ms p(95)=1.01s http_reqs.....: 616 0.96874/s iteration_duration.....: avg=1.97s min=1.36s med=1.84s max=5.22s p(90)=2.56s p(95)=2.98s iterations.....: 300 4.96477/s vus.....: 4 min=4 max=10 vus_max.....: 10 min=10 max=10  Running (1m01.8s), 00/10 VUs, 308 complete and 0 interrupted iterations default [*****] 10 VUs 1m0s </pre>	El sistema responde a las solicitudes con promedios que no superan los 500ms.
------	--	------------	---	---

Nota: Se usan 10 usuarios ya que el proyecto cuenta con 14 especialistas. Además, se determinó la adquisición de 5 laptops durante la compra de recursos materiales. (5)

### 3.3.3. Pruebas de seguridad

Tabla 37. Pruebas de seguridad

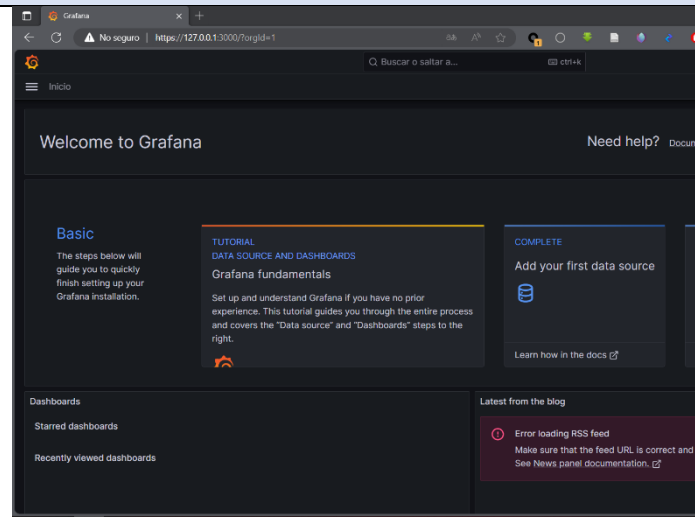
ID de Escenario	Escenario	Herramienta	Respuesta de la herramienta	Resultado
ES 1	Comprobar uso de protocolo HTTPS	Postman	 <pre> GET https://127.0.1:3000 {   "execution_time": 1,   "total_time": 1 } </pre>	Se puede establecer una conexión mediante protocolo HTTPS. El sistema

				tiene el protocolo HTTPS habilitado.
ES 2	Comprobar uso de UUID	Navegadores		Se puede comprobar que los dashboards emplean UUID. Sin embargo, los usuarios utilizan un entero como ID.
ES 3	Escanear vulnerabilidades de seguridad	OSWAP ZAP		Saltan varias alertas de seguridad. Entre ellas, las de

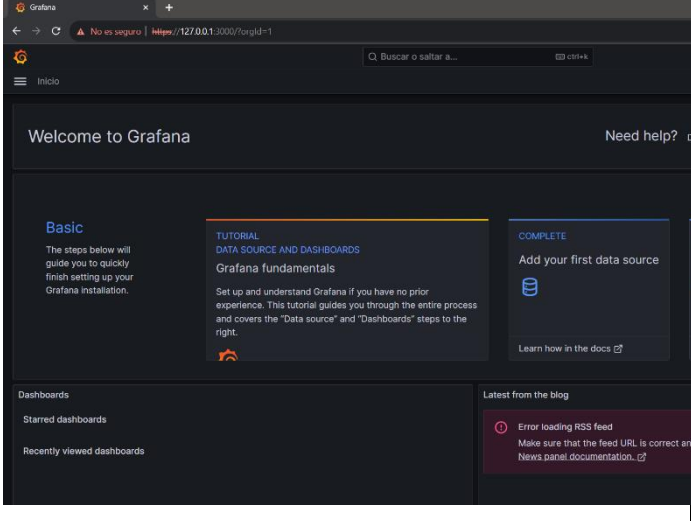
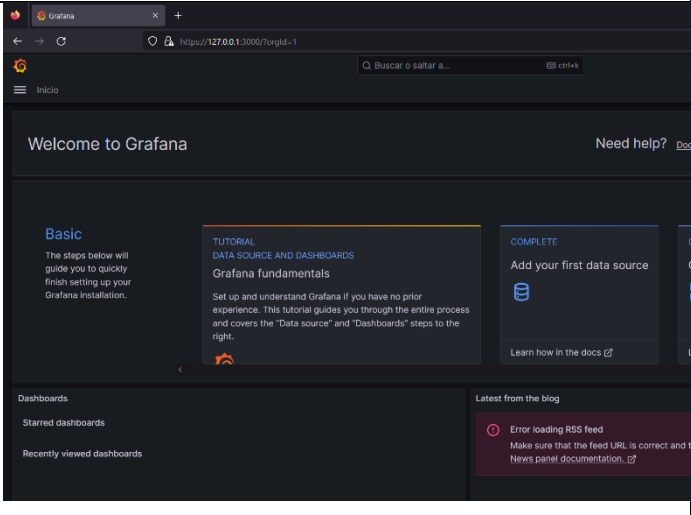
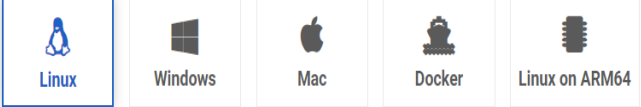
				prioridad alta y media constituyen las de mayor riesgo.
--	--	--	--	---

### 3.3.4. Pruebas de portabilidad

**Tabla 38.** Pruebas de portabilidad

ID de Escenario	Escenario	Herramienta	Respuesta de la herramienta	Resultado
EP 1	Comprobar portabilidad en Microsoft Edge	Navegadores		Disponible para Edge



<p>EP 2</p>	<p>Comprobar portabilidad en Google Chrome</p>	<p>Navegadores</p>		<p>Disponible para Chrome</p>
<p>EP 3</p>	<p>Comprobar portabilidad en Mozilla Firefox</p>	<p>Navegadores</p>		<p>Disponible para Firefox</p>
<p>EP 4</p>	<p>Comprobar portabilidad en sistemas operativos</p>	<p>Navegadores</p>		<p>Según su página oficial de descarga, Grafana cuenta con versiones de</p>

				descarga para estos sistemas operativos.
--	--	--	--	--

### Conclusiones parciales

Con el uso de las descripciones de las páginas principales que conforman el sistema, se describe su funcionamiento. El diagrama de despliegue determina el comportamiento del sistema una vez desplegado y de los nodos que lo conforman, destacando el empleo de la arquitectura cliente-servidor. Mediante el uso de pruebas de aceptación, promovidas por la metodología XP como uno de los tipos de prueba a emplear, se llega a la conclusión que el sistema cumple con todas las especificaciones funcionales determinadas. Las pruebas de aceptación resultan satisfactorias. Con el empleo de herramientas para medir el rendimiento, se comprueba que el sistema soporta cargas de trabajo considerable y su rendimiento no se ve mermado. Sin embargo, con las pruebas de seguridad no sucede lo mismo. Con el uso de la herramienta OWASP ZAP para hacer ataques de penetración, se determina que el sistema tiene brechas de seguridad que pudieran comprometerlo. Mediante el uso de los navegadores se pudo comprobar que los usuarios del sistema no son identificados con UUID.

## Conclusiones generales

Este trabajo ha logrado diseñar e implementar un sistema de visualización utilizando Grafana que permite la presentación efectiva de una amplia gama de indicadores relevantes para la vigilancia tecnológica. A través de la implementación de este sistema, se ha logrado una representación visual efectiva de tendencias y patrones en los indicadores de vigilancia tecnológica. Esto es esencial para identificar oportunidades, amenazas y posibles puntos de inflexión en el panorama tecnológico.

Este sistema se considera una herramienta crucial para fomentar la innovación al proporcionar a los tomadores de decisiones una comprensión detallada del panorama tecnológico actual, permitiéndoles anticipar y adaptarse a cambios significativos en el mercado.

Se identificó el potencial de este sistema para influir en la agilidad de la toma de decisiones estratégicas, así como para mejorar la capacidad de la organización para identificar tendencias, oportunidades y amenazas en el entorno tecnológico.

Los módulos constituyen la base del sistema y proporcionan una amplia gama de capacidades para la visualización y análisis de datos, así como la creación de paneles de control personalizados para monitorizar sistemas y aplicaciones. La modularidad y la flexibilidad le permite adaptarse a una gran variedad de entornos y casos de uso, convirtiéndolo en una herramienta potente para la visualización de datos y el monitoreo. Ofrece flexibilidad en la forma en que se pueden visualizar y analizar los datos. Esto es crucial en un entorno de vigilancia tecnológica, donde los requisitos y las fuentes de datos pueden variar ampliamente. Los usuarios pueden interactuar con los datos y personalizar las visualizaciones según sus necesidades específicas. Esto es fundamental para la adaptación a diferentes casos de uso en vigilancia tecnológica.

A través del desarrollo y despliegue de este sistema, se ha profundizado en la comprensión de la importancia de la visualización de datos en el proceso de vigilancia tecnológica, destacando su valor para la toma de decisiones.

## Recomendaciones

Se sugiere la implementación de programas de formación y capacitación para los usuarios finales del sistema de visualización. Esto garantizará que comprendan plenamente las capacidades del sistema y puedan utilizarlo para maximizar su eficacia en la vigilancia tecnológica.

Se recomienda un proceso continuo de desarrollo y evaluación del sistema de visualización. Esto incluiría la consideración de actualizaciones, nuevas características y evaluaciones periódicas para garantizar que el sistema siga siendo relevante y efectivo en un entorno en constante evolución.

Se anima a explorar plenamente todas las capacidades de Grafana, incluyendo la configuración avanzada de visualizaciones, alertas personalizadas y la integración con nuevas fuentes de datos para enriquecer la vigilancia tecnológica.

Considerando la importancia de la visualización de datos para la vigilancia tecnológica, se recomienda continuar perfeccionando este sistema, explorando enfoques adicionales para la presentación de datos y la integración de análisis predictivos para mejorar aún más la toma de decisiones basada en datos.

Aunque este sistema ofrece herramientas poderosas para la visualización de datos, también se identificaron desafíos relacionados con la selección y presentación eficiente de indicadores relevantes, lo que puede ser un área de enfoque para futuras mejoras.

Se sugiere la publicación de los resultados y aprendizajes derivados de la investigación en revistas académicas especializadas o la presentación en conferencias del ámbito de la vigilancia tecnológica y la visualización de datos.

## Referencias bibliográficas

1. ¿Qué es la visualización de datos? | Oracle España. Online. 2023. [Accessed 20 September 2023]. Available from: <https://www.oracle.com/es/business-analytics/what-is-data-visualization/>
2. ¿Qué es la visualización de datos? | Tableau. Online. 2023. [Accessed 19 September 2023]. Available from: <https://www.tableau.com/es-mx/learn/articles/data-visualization>
3. ¿Qué es la visualización de datos? | IBM. Online. 2023. [Accessed 19 September 2023]. Available from: <https://www.ibm.com/es-es/topics/data-visualization>
4. UNE 166006:2018 Gestión de la I+D+i. Online. 2023. [Accessed 22 October 2023]. Available from: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0059973>
5. CORREA, Omar. *Vigilancia Científico Técnica en TIC con uso intensivo de tecnologías*. 2023.
6. Norma Cubana NC 1308:2019. Online. 2023. [Accessed 22 October 2023]. Available from: [https://www.presidencia.gob.cu/media/filer/public/2022/10/11/cuba\\_nc\\_1308.pdf](https://www.presidencia.gob.cu/media/filer/public/2022/10/11/cuba_nc_1308.pdf)
7. MARTÍNEZ ALBARRACÍN, C., BOLÍVAR SALGADO, W. and GUTIÉRREZ SALGADO, N. Information source observatories: Concepts and theoretical review. *Journal of business and entrepreneurial studies*. Online. 2021. Vol. 5. Available from: <https://www.redalyc.org/articulo.oa?id=573669774002>
8. SARMIENTO REYES, Y., DELGADO FERNÁNDEZ, M. and INFANTE ABREU, M. Observatorios: clasificación y concepción en el contexto iberoamericano. *Revista Cubana de Información en Ciencias de la Salud*. Online. 2019. Vol. 30. Available from: <https://www.redalyc.org/articulo.oa?id=377665578007>
9. ¿Qué es el análisis de datos? - Explicación del análisis de datos - AWS. *Amazon Web Services, Inc.* Online. 2023. [Accessed 20 September 2023]. Available from: <https://aws.amazon.com/es/what-is/data-analytics/>

10. ¿Qué es la visualización de datos? - Explicación de la visualización de datos - AWS. *Amazon Web Services, Inc.* Online. 2023. [Accessed 15 September 2023]. Available from: <https://aws.amazon.com/es/what-is/data-visualization/>
11. VIERA, Yaquelín Córdova, BORREGO, Jennifer Martínez and VIERA, Elizabet Córdova. Propuesta de metodología para el diseño de dashboard. *Revista cubana de transformación digital*. 2021. Vol. 2, no. 3, p. 56–76.
12. FEW, Stephen. *Information dashboard design: The effective visual communication of data*. . O'Reilly Media, Inc., 2006. ISBN 0-596-10016-7.
13. Qué es un panel de datos | Microsoft Power BI. Online. 2023. [Accessed 22 October 2023]. Available from: <https://powerbi.microsoft.com/es-es/data-dashboards/>
14. Dashboards de BI | Qué debe saber | Tableau. Online. 2023. [Accessed 21 September 2023]. Available from: <https://www.tableau.com/es-es/learn/articles/business-intelligence/bi-dashboards>
15. HARLOW, Jason. What is a Key Performance Indicator (KPI)? *KPI.org*. Online. 2023. [Accessed 22 October 2023]. Available from: <https://www.kpi.org/kpi-basics/>
16. Qué es un KPI y cómo aprovecharlo para su negocio. Online. 2023. [Accessed 22 October 2023]. Available from: <https://www.tableau.com/es-mx/learn/articles/what-is-kpi>
17. Fusion Middleware Guía del Usuario de Oracle Business Intelligence Enterprise Edition. Online. 2023. [Accessed 22 October 2023]. Available from: <https://docs.oracle.com/middleware/12212help/biee/es/BIEUG/GUID-925560F8-1EEE-4887-BA4D-B8C5A7482B76.htm#BIEUG1228>
18. Guía de Indicadores de Ciencia, Tecnología e Innovación — OVTT. Online. 6 May 2020. [Accessed 22 October 2023]. Available from: <https://www.ovtt.org/guias/guia-practica-de-indicadores-de-ciencia-y-tecnologia/>
19. TIOBE Index. *TIOBE*. Online. 2023. [Accessed 15 September 2023]. Available from: <https://www.tiobe.com/tiobe-index/>

20. PYPL Popularity of Programming Language index. Online. 2023. [Accessed 22 October 2023]. Available from: <https://pypl.github.io/PYPL.html>
21. SOMMERVILLE, Ian. *Ingeniería del software*. . Pearson educación, 2011. ISBN 84-7829-074-5.
22. Grafana documentation. *Grafana Labs*. Online. 2023. [Accessed 15 September 2023]. Available from: <https://grafana.com/docs/grafana/latest/>
23. Visual Paradigm User's Guides. Online. 2023. [Accessed 15 September 2023]. Available from: <https://www.visual-paradigm.com/support/documents/>
24. Sparx Systems - UML Resources. Online. 2023. [Accessed 15 September 2023]. Available from: [https://sparxsystems.com/platforms/uml\\_resources.html](https://sparxsystems.com/platforms/uml_resources.html)
25. Documentation for Visual Studio Code. Online. 2023. [Accessed 15 September 2023]. Available from: <https://code.visualstudio.com/docs>
26. Welcome to Python.org. *Python.org*. Online. 2023. [Accessed 15 September 2023]. Available from: <https://www.python.org/doc/>
27. MCKINNEY, Wes. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. . O'Reilly Media, Inc., 2012. ISBN 1-4493-2361-8.
28. pandas documentation — pandas 2.1.0 documentation. Online. 2023. [Accessed 15 September 2023]. Available from: <https://pandas.pydata.org/docs/>
29. pandas - Python Data Analysis Library. Online. 2023. [Accessed 15 September 2023]. Available from: <https://pandas.pydata.org/about/>
30. CHAUHAN, Anjali. A review on various aspects of MongoDB databases. *International Journal of Engineering Research & Technology (IJERT)*. 2019. Vol. 8, no. 05, p. 90–92.
31. What is MongoDB? — MongoDB Manual. Online. 2023. [Accessed 15 September 2023]. Available from: <https://www.mongodb.com/docs/manual/>
32. ZAP – Getting Started. Online. 2023. [Accessed 31 October 2023]. Available from: <https://www.zaproxy.org/getting-started/>

33. k6 Documentation. Online. 2023. [Accessed 31 October 2023]. Available from: <https://k6.io/docs>
34. What is Postman? Postman API Platform. *Postman API Platform*. Online. 2023. [Accessed 1 November 2023]. Available from: <https://www.postman.com/product/what-is-postman/>
35. SMITH, Steve. Información general de ASP.NET Core MVC. Online. 3 August 2023. [Accessed 21 September 2023]. Available from: <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-7.0>
36. RODRÍGUEZ-CRUZ, Yunier and PINTO, María. Requerimientos informacionales para la toma de decisiones estratégicas en organizaciones de información. *Transinformação*. 2017. Vol. 29, no. 2, p. 175–189.
37. ASANA. Historias de usuario: 3 ejemplos para generar valor para el usuario [2022] • Asana. *Asana*. Online. 2023. [Accessed 15 November 2023]. Available from: <https://asana.com/es/resources/user-stories>
38. PRESSMAN, Roger S. Ingeniería del software. . 2010.