

ARTÍCULO ORIGINAL

 <https://doi.org/10.5281/zenodo.5545998>

# Propuesta de metodología para el diseño de *dashboard*

*Proposed Methodology for Dashboard Design*

*Yaquelin Córdova Viera*

*ycordovav@uci.cu* • <https://orcid.org/0000-0002-8370-0488>

*Jennifer Martínez Borrego*

*jmartinezb@uci.cu* • <https://orcid.org/0000-0002-9401-9444>

*Elizabet Córdova Viera*

*ecviera@uci.cu* • <https://orcid.org/0000-0002-7868-6570>

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, CUBA

Recibido: 2021-06-02 • Aceptado: 2021-07-12

## RESUMEN

Los *dashboards* son herramientas que permiten compartir, agrupar, centralizar y proporcionar una visualización gráfica de la información relevante de una organización, facilitando la toma de decisiones. Para el desarrollo de *dashboard* existen metodologías que permiten monitorear y dar seguimiento a indicadores claves, creadas por Noetix Corporation, R. Edis, y M. Jusko. Estas metodologías proporcionan una guía para llevar a cabo la implementación exitosa de un *dashboard* en una organización, las cuales fueron tomadas en cuenta por los autores. La presente investigación tiene como objetivo desarrollar una metodología (basada en las mejores prácticas evidenciadas en las metodologías estudiadas) que permita diseñar *dashboard* para los proyectos del Centro de Representación y Análisis de Datos (CREAD) de la Universidad de las Ciencias Informáticas (UCI). La metodología desarrollada se estructuró en cinco fases, también se incluyeron pasos específicos dentro de cada fase que permitieron, la organización y actualización de los datos, la selección de los tipos de gráfico, la configuración de umbrales y alertas. Para validar la metodología se aplicó en el diseño de un *dashboard* para el proyecto SIGE v3.0 del Ministerio de Educación Superior (MES). La utilización del *dashboard* permitió a los responsables del seguimiento de indicadores claves tener una perspectiva clara de la situación

actual de la organización, ya que los indicadores a monitorear son específicos del MES y resultan determinantes a la hora de alcanzar objetivos y tomar decisiones.

**PALABRAS CLAVE:** *dashboard*, indicadores claves, SIGE v3.0.

## ABSTRACT

*Dashboards are tools that allow sharing, grouping, centralizing and providing a graphic visualization of the relevant information of an organization, facilitating decision making. For the development of dashboards there are methodologies that allow monitoring and follow-up of key indicators, created by Noetix Corporation, R. Edis, and M. Jusko. These methodologies provide a guide to carry out the successful implementation of a dashboard in an organization, which were taken into account by the authors. The present research aims to develop a methodology (based on the best practices evidenced in the methodologies studied) that allows designing dashboard for the projects of the Centro de Representación y Análisis de datos (CREAD) of the Universidad de las Ciencias Informáticas (UCI). The developed methodology was structured in five phases, also specific steps were included within each phase that allowed the organization and updating of data, the selection of chart types, the configuration of thresholds and alerts. To validate the methodology, it was applied in the design of a dashboard for the SIGE v3.0 project of the Ministerio de Educación Superior (MES). The use of the dashboard allowed those responsible for monitoring key indicators to have a clear perspective of the current situation of the organization, since the indicators to be monitored are specific to the MES and are decisive when it comes to achieving objectives and making decisions.*

**KEYWORDS:** *dashboard*, key indicators, SIGE v3.0.

## INTRODUCCIÓN

El volumen y variedad de los datos de las organizaciones ha crecido exponencialmente en las últimas décadas, de tal forma que las empresas e instituciones en el mundo se han visto en ocasiones abarrotadas de datos históricos que no aprovechan al máximo. Estos datos, bien tratados y analizados pueden ser convertidos en información y reportar beneficios a las organizaciones, lo que ha hecho necesario la creación de tecnologías que permitan su organización y procesamiento, posibilitando extraer conocimiento útil de la información

almacenada (Quesada, 2008). Esto se puede lograr con el uso de la Inteligencia de negocios o *Business Intelligence* (BI), término definido por Davenport<sup>1</sup> y Harris<sup>2</sup> en 2007, los cuales afirmaban que “la Inteligencia de Negocios consiste en la obtención, administración y reporte de los datos orientados a la toma de decisiones, y las técnicas analíticas y procesos computarizados que se usan para el análisis de la misma” (Ferrer, 2019).

Las herramientas de BI permiten realizar consultas y análisis sobre los datos que se encuentran en una base de datos; con soluciones basadas, por ejemplo: almacenes de datos, mercados de datos y *dashboard*. Los *dashboard* o cuadro de mando (CM), muestran información relevante, presentada en una manera que permite hacer un seguimiento de lo que está ocurriendo en un instante de tiempo. Para servir a su propósito y maximizar sus prestaciones, deben mostrar abundante información en forma de elementos visuales, en una pequeña cantidad de espacio, de manera que se comuniquen con claridad e inmediatez el comportamiento de la información para un posterior análisis. Esto requiere un diseño que se nutra y aproveche el poder de la percepción visual para lograr el procesamiento de grandes cúmulos de información (Few, 2007).

En Cuba a principios del año 2000 comenzaron a implementarse en las empresas cubanas diferentes sistemas informáticos para la administración de las empresas. En el año 2003 la Agencia de inspección, ajuste de averías y otros servicios conexos (Intermar Cienfuegos) tenía un *dashboard* para la gestión empresarial. Posteriormente, en el año 2004 la Empresa de Servicios Informáticos Especializados en Turismo (GET Varadero) y Empresa de Servicios Especializados de Protección, S.A. (SEPSA Cienfuegos), y luego, en los años 2005 y 2007 respectivamente, la Empresa de Servicios Técnicos de Defectoscopía y Soldadura (CENEX) y Empresa de Diseño Ciudad Habana (DCH), habían incorporado a su sistema empresarial *dashboard* para la toma de decisiones (Soler González, 2009a) (Soler González, 2009b).

Para el desarrollo de *dashboard*, existen metodologías que permiten diseñarlos basados en elementos comunes para conseguir el seguimiento y evolución de indicadores, en el año 2004 la empresa Noetix<sup>3</sup> elaboró la metodología “Desarrollo e implementación del *dashboard*-Metodología para el éxito”. Luego en el año 2016 Robert Edis propuso el “Uso de metodologías ágiles con implementación de *dashboard* de inteligencia de negocios” y M. Jusko, “Una metodología para crear *dashboard*” en el año 2017. Estas metodologías proporcionan una guía de los diferentes puntos de vista de los autores para llevar a cabo la implementación exitosa de un *dashboard* en una organización, de tal manera que, cada una de ellas aportó con especificaciones que fueron tomadas en cuenta para la realización de esta metodología.

La metodología propuesta en este artículo proporciona una serie de pasos que se consideren válidos y pueden ser puestos en práctica, y, es de utilidad cuando al momento de desarrollar e implementar un *dashboard* se cuenta con limitaciones como el tiempo.

---

<sup>1</sup> Thomas H. Davenport es un autor estadounidense académico, especializado en el análisis y la innovación de los procesos de negocio.

<sup>2</sup> J. G. Harris es un profesor del Departamento de Gestión de la Información de la Escuela de Gestión.

<sup>3</sup> Noetix: empresa fundada en 1994 por Redmond Washington en Estados Unidos, especializada en software de inteligencia de negocios.

El excesivo uso de información, así como obviar normas de usabilidad al momento de desarrollar un *dashboard* pueden ocasionar que la información que se desea transmitir no sea la correcta, y, de esta manera las fallas de diseño se vuelven una carga para las personas encargadas de darle seguimiento a la información.

La utilización de un *dashboard* adecuado permitirá a los responsables del seguimiento de indicadores claves tener una perspectiva clara de la situación actual de una organización, evitando las distracciones que una herramienta incompleta pueda proporcionar, ya que los indicadores a monitorear son específicos y pueden resultar determinantes a la hora de alcanzar objetivos en la organización.

La implementación de un *dashboard* requiere de una metodología que considere todos los aspectos a ser desarrollados dentro del ciclo de vida del *software*.

Se propone como objetivo general: Desarrollar una metodología que permita diseñar e implementar de manera exitosa *dashboard* para los proyectos del Centro de Representación y Análisis de Datos (CREAD) de la UCI.

## MATERIALES Y MÉTODOS

### Métodos Teóricos:

- Histórico-lógico
- Análisis y síntesis

### Métodos Empíricos:

- Entrevista
- Observación participativa

## ESTADO DEL ARTE

Entre las metodologías para el diseño de *dashboard* se encuentran: Desarrollo e implementación del *dashboard*-Metodología para el éxito de la empresa Noetix (2004), VROps-Una metodología para crear *dashboards* de M. Jusko (2017), y Metodologías ágiles con implementación de *dashboard* de inteligencia de negocios de Robert Edis (2016).

### Desarrollo e implementación del dashboard-Metodología para el éxito

Es una metodología diseñada por la empresa Noetix en el año 2004, esta metodología describe el proceso necesario para planificar, diseñar, construir e implementar correctamente un *dashboard*, independientemente de la tecnología que se escoja. Cuenta con seis fases: planificación, recopilación de requisitos, diseño, construcción y validación, despliegue y mantenimiento (Orts, 2004).

- **Planificación:** la fase de planificación es donde todo comienza, deben quedar claros los miembros del equipo del proyecto, sus funciones definidas e identificar los indicadores que se van a analizar.
- **Recopilación de requisitos:** una vez que se haya definido la planificación, comienza el proceso de recopilación de requisitos. Se deben entrevistar a los principales interesados

para determinar sus necesidades y expectativas para con el *dashboard*, estas necesidades deben asignarse a los indicadores previamente identificados.

- **Diseño:** luego de definir los requisitos para crear el *dashboard*, los aspectos principales del diseño deben ser completados (confirmar la fuente de datos, estructura de los datos que se mostrarán, consultas SQL para acceder a los indicadores).
- **Construcción y validación:** en esta fase se producen cuatro tareas (implementación de la interfaz, implementación de las consultas, programación; actualización, seguridad y la validación del *dashboard*).
- **Despliegue:** una vez que el *dashboard* ha sido construido y probado, se lleva a producción para así implementar los requisitos de seguridad y se realiza la integración en un entorno de red corporativa (*framework*, intranets para el acceso de socios y clientes).
- **Mantenimiento:** en esta fase se debe tomar medidas para proporcionar un mantenimiento continuo de la aplicación ya que con el tiempo las necesidades del proyecto podrían variar.

### VROps-Una metodología para crear dashboards

Esta metodología propuesta por Jusko en el año 2017, se centra en las interacciones de los datos que tienen los usuarios con las métricas de los KPI que se muestran en el *dashboard* a través del uso de elementos denominados *widgets*, y en el diseño visual de los objetivos. Además, proporciona una pequeña guía sobre la convención de uso de técnicas de diseño y usabilidad relacionadas al usuario final.

Propone siete fases: definir el objetivo del *dashboard*, planificar un flujo de trabajo, conocer sus datos, elegir los *widgets*, planificar interacciones, pruebas de manejo, refinamiento y mantenimiento (Jusko, 2017).

- **Definir el objetivo del dashboard:** en esta fase se define un objetivo de negocio, la consecución de este objetivo le dará al *dashboard* el propósito y el valor.
- **Planificar un flujo de trabajo:** cada flujo de trabajo debe ser fácilmente repetible para diferentes usuarios. Entre el inicio y el final del flujo de trabajo, todos los objetos, métricas, relaciones y capacidades del producto deben tomarse en consideración.
- **Conocer sus datos:** antes de crear contenido para un *dashboard*, debe completarse un análisis exhaustivo de los datos del entorno para descubrir objetos y métricas que se relacionan con el objetivo de negocio y como dichos elementos están asociados entre si usando relaciones.
- **Elegir los widgets:** en esta fase se comienza el proceso de selección de elementos gráficos. Los *widgets* se utilizan para presentar información a los usuarios, permitir la interacción y el análisis de los datos y, por último, formar los bloques de construcción del *dashboard*.
- **Planificar interacciones:** las interacciones permiten que los datos se comuniquen entre los *widgets* y benefician la reducción de *widgets* redundantes y la información estática en los *dashboard*.

- **Pruebas de manejo:** en esta fase se implementa el *dashboard* para comprobar si realmente resuelve los objetivos fijados. Se realizan revisiones iterativas ya que las expectativas y los requisitos del *dashboard* pueden cambiar.
- **Refinamiento y mantenimiento:** en esta fase se actualiza y se configura la información contenida en el *dashboard*.

### Uso de metodologías ágiles con implementación de dashboard de inteligencia de negocios

Es una metodología creada por Robert Edis en el año 2016, basada principalmente en el enfoque de metodologías ágiles a través de Scrum, que proporciona las herramientas de planificación para la implementación de un *dashboard*, basado en los sprints del marco de trabajo scrum.

El enfoque ágil está diseñado para acelerar el desarrollo de entregas de *software* utilizando una combinación de prototipos y retroalimentación oportuna de los representantes de los usuarios finales, el trabajo se divide en pequeños lapsos de tiempos llamados sprints.

Propone seis fases: planear el sprint, taller de diseño, prototipado, pruebas, implementación, sprints especiales (Edis, 2016).

- **Planear el sprint:** en esta fase se decide la duración, las entregas, las responsabilidades y la logística, como los lugares del taller y reunión, las horas y los materiales que se emplearán en el diseño del *dashboard*.
- **Taller de diseño:** en esta fase se especifican los requisitos de negocio para derivar los requisitos de diseño funcional, los resultados de este paso incluyen una maqueta estática para visualizar cómo será el *dashboard* y debe contener todos los elementos requeridos para la entrega final.
- **Prototipado:** en esta fase se diseñan los prototipos que contienen datos representativos y variados que se van a encontrar en la producción.
- **Pruebas:** se realizan tipos de pruebas para comprobar si el funcionamiento del *dashboard* creado cumple con las funcionalidades requeridas. Recomienda niveles de prueba tales como: pruebas unitarias, pruebas de integración del sistema, pruebas de aceptación de usuario que pueden ser incluidas en los sprint.
- **Implementación:** en esta fase se realizan las entregas de sprint que deben ser implementadas a los usuarios finales tan pronto como sea posible.
- **Sprints especiales:** si existen cambios en el producto final deben ser manejados usando procedimientos de control de cambios apropiado.

Los aportes de cada metodología son los siguientes (Martínez-Robalino, 2017):

- La metodología “Desarrollo e implementación del *dashboard*-Metodología para el éxito” se centra en la generación de contenido relevante, basado en las configuraciones únicas de los clientes, considerando que la mayor parte de la automatización se produce dentro de los primeros días de implementación y puede reducir drásticamente el tiempo y el esfuerzo del proyecto a pocas semanas.

- La metodología “VROps- Metodología para crear *dashboard*” se centra en las interacciones de los datos con las métricas de los KPI, y en el diseño visual de los objetivos.
- El “Uso de metodologías ágiles con implementación de *dashboard* de inteligencia de negocios” proporciona las herramientas de planificación para la implementación de un *dashboard*, basado en los sprints del marco de trabajo de scrum.

Una vez que han sido analizadas estas metodologías se identificó que no todas contemplan el ciclo de desarrollo de *software* completo, en ninguna de las tres metodologías se propone la fase de seguridad de los datos. Las dos primeras metodologías mencionadas incluyen la fase de pruebas, pero no recomiendan los niveles y tipos de pruebas a realizar. También la documentación de las metodologías es escasa.

Por los elementos antes mencionados se considera apropiado desarrollar una metodología para el diseño de *dashboard*, basada en las mejores prácticas evidenciadas en las metodologías estudiadas. En la metodología propuesta (ver Tabla 1) se integrarán las fases de la metodología de Noetix (menos las fases “Despliegue” y “Mantenimiento”), en Noetix se plantea en una sola fase “Construcción y Validación” del *dashboard*, por la importancia de esta fase se han dividido en dos fases. Al mismo tiempo, la seguridad de los datos será tratada como un paso dentro la fase cuatro “Construcción del *dashboard*”. En la fase cinco “Validación del *dashboard*” se detallará la estrategia de pruebas a seguir. Además, la fase “Conocer sus datos” de la metodología de Jusko se adicionará como un paso (con el nombre “Análisis de la fuente de datos”) en la fase tres “Diseño del *dashboard*”. También es necesario destacar que se incluirán pasos específicos dentro de cada fase, que orienten a las personas encargadas de diseñar el *dashboard*.

**Tabla 1. Fases de la metodología propuesta. Fuente: Elaboración propia.**

Fases de la metodología de Noetix	Fases de la metodología propuesta	Breve descripción de las fases de la metodología propuesta	Métodos o herramientas que se proponen utilizar
1. Planificación	1. Planificación	Se inicia el proceso del diseño del dashboard, describiendo las características de la organización y escogiendo a los miembros del equipo del proyecto.	-Diagrama de Gantt
2. Recopilación de requisitos	2. Recopilación de requisitos	Se definen los requisitos de información, funcionales y no funcionales.	-Tormenta de ideas -Entrevista estructurada
3. Diseño	3. Diseño del dashboard	Se realiza un análisis de la fuente de datos, se hace una selección de estos y se define su estructura y almacenamiento.	-DataCleaner v7.1 -InfluxDB v1.7.1
4. Construcción y validación	4. Construcción del dashboard	Se evalúan y definen qué tipos de gráficos representan mejor los datos que se van a mostrar, se configuran las alertas y las notificaciones.	-Grafana v6.6.1
	5. Validación del dashboard	Se define una estrategia de pruebas para asegurar que el dashboard cumple con los requisitos y especificaciones requeridas.	-Apache JMeter v2.11 -Acunetix Web Vulnerability Scanner
5. Despliegue			
6. Mantenimiento			

## HERRAMIENTAS Y TECNOLOGÍAS

Una vez analizadas las principales características que poseen los sistemas gestores de bases de datos (SGBD) Elasticsearch (DB-Engines, 2020c) (Elasticsearch, 2020) e InfluxDB (DB-Engines, 2020d) (Influxdb, 2020), los autores proponen para el desarrollo de *dashboard* utilizar InfluxDB en su versión 1.7.1, ya que es una de las bases de datos de series de tiempo más manejadas en la actualidad, según los rankings DB-Engines que se han consultado (DB-Engines, 2020a) (DB-Engines, 2020b). Soporta un mayor número de lenguajes de programación con respecto al resto y puede obtener datos mediante métodos de acceso HTTP, API y JSON (DB-Engines, 2020e). Actualmente es complicado encontrar información técnica y bibliográfica de las bases de datos temporales, más allá de las que ofrecen los sitios web oficiales de cada una, siendo InfluxDB de las que más información se puede llegar a encontrar en la red mediante foros y artículos.

Existen herramientas de visualización para ayudar al análisis de la información que permiten descubrir y comprender la lógica que se encuentra detrás de un conjunto de información, así como para compartir esta interpretación con otras personas desde un punto de vista objetivo (Pérez-Montoro, 2017). Entre las herramientas para la visualización de indicadores se encuentran *Kibana* y *Grafana*, las cuales permiten utilizar como SGBD Elasticsearch e InfluxDB respectivamente. Una vez realizado un análisis de las herramientas para la visualización de indicadores (Morales, 2016), ambas herramientas poseen ventajas similares, sin embargo, *Kibana* es dependiente de *Elasticsearch* mientras que *Grafana* es independiente de la fuente de datos a utilizar. En la bibliografía consultada (Ballesteros, 2017) (Nazco, 2018) (Sainz, 2020) se evidencia que un número elevado de *dashboard* diseñados en *Grafana* utilizan InfluxDB como servidor de base de datos, lo que demuestra que ambas herramientas se comunican correctamente. *Grafana* proporciona privacidad de la información a través de la gestión de roles y puede trabajar con múltiples bases de datos. Además, permite combinar datos de diferentes fuentes en un solo tablero. Los autores de esta investigación sugieren utilizar la herramienta *Grafana* en su versión 6.6.1.

## RESULTADOS Y DISCUSIÓN

En la figura 1 se muestran las cinco fases (con sus pasos específicos) que componen la metodología propuesta.

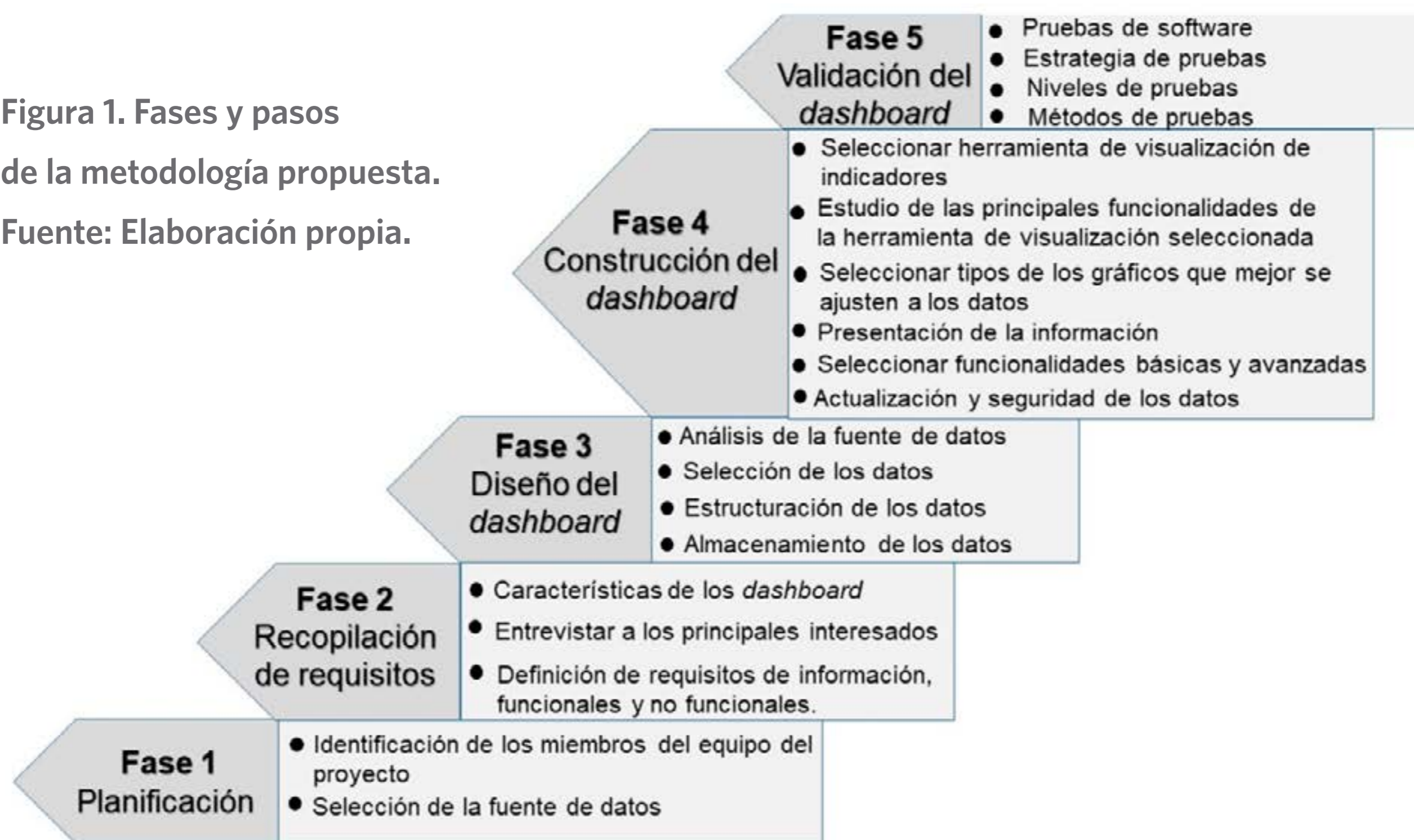
### FASE 1: PLANIFICACIÓN

La fase de planificación es la que inicia el proceso del diseño del *dashboard*, se exponen las principales características y funciones de la organización, además de los problemas existentes que conllevan al diseño del *dashboard*.

En este punto los miembros del equipo del proyecto deben ser identificados y sus funciones claramente definidas. Además, se selecciona la fuente de datos existente, de donde se extraerán los datos y cómo se realizará el acceso a estos datos.



Figura 1. Fases y pasos de la metodología propuesta.  
Fuente: Elaboración propia.



## FASE 2: RECOPIACIÓN DE REQUISITOS

Una vez que se ha definido la planificación, comienza el proceso de recopilación de requisitos. En este punto se deben definir las características del *dashboard* para que sea útil en la organización y se propone además realizar entrevistas a los principales interesados para determinar las necesidades y expectativas con respecto al *dashboard*.

### Paso 2.1 Características de los *dashboard*

El equipo de proyecto debe evaluar las características de los *dashboard* teniendo en cuenta las que ofrece *Juice Analytics* (ver tabla 2) y en la entrevista a realizar a los principales interesados (próximo paso) seleccionar las que serán útiles para el *dashboard* a diseñar para la organización.

Tabla 2. Características de los *dashboard* (Juice, 2009).

Características				
Alcance	Ancho: Se muestra información sobre toda la organización.	Específico: Se centra en una función específica, proceso, producto.		
Rol comercial	Estratégico: Proporciona una visión de alto nivel, amplia y a largo plazo del rendimiento.	Operacional: Proporciona una visión enfocada, a corto plazo y táctica del rendimiento.		
Tiempo	Histórico: Realiza una mirada hacia atrás para seguir las tendencias.	Instantánea: Se muestra el rendimiento en un solo punto en el tiempo.	Tiempo real: Se monitorea la actividad mientras sucede.	Profético: Se utiliza el rendimiento pasado para predecir el rendimiento futuro.
	Talla única: La información es presentada como una única vista para todos los usuarios.	Personalizable: Funcionalidad que permite a los usuarios crear una vista que refleje sus necesidades.		
Nivel de detalle	Alto: Son presentados solo los números de nivel superior más críticos.	Profundo: Se proporciona la capacidad de profundizar en números detallados para ganar más contexto.		
Punto de vista	Explicativo: El <i>dashboard</i> le dice explícitamente al usuario qué significan los datos y qué hacer al respecto.	Exploratorio: El usuario tiene libertad para interpretar los resultados según se muestran.		

## **Paso 2.2 Entrevistar a los principales interesados**

Se recomienda utilizar preguntas abiertas en las entrevistas, donde los entrevistados puedan elaborar y dar detalles, más allá de simplemente responder “sí” o “no”. Durante la entrevista debe garantizarse que la información recogida no solo vaya enfocada a las necesidades que posteriormente serán transformadas en requisitos funcionales, sino que además se deben definir las necesidades que representarán los requisitos no funcionales y los datos que serán analizados en cuanto a métricas en los requisitos de información.

## **Paso 2.3 Definición de requisitos de información, funcionales y no funcionales**

### **Requisitos de información**

Los RI en los *dashboard* están asociados a los indicadores que se seleccionen para ser analizados y el sistema debe permitir mostrar su comportamiento en el tiempo.

### **Requisitos funcionales**

Los RF en los *dashboard* deben estar orientados a las necesidades de los usuarios finales, estos requisitos pueden ser implementados o garantizados por la herramienta de visualización que se escoja, la cual puede brindar varias de las funcionalidades identificadas como RF.

### **Requisitos no funcionales**

En los *dashboard* los RNF describen las características que deben poseer las computadoras clientes, así como las características del servidor de base de datos y del servidor de la herramienta de visualización.

Luego de elaborada las preguntas a realizar a los principales interesados, desarrollar las entrevistas, y revisar y documentar el resultado de las mismas, se debe proceder a identificar los RI, RF y los RNF que tendrá el *dashboard*.

## **FASE 3: DISEÑO DEL DASHBOARD**

Luego de recopilar los requisitos que deben estar presentes en el sistema, se puede comenzar a diseñar el *dashboard*, para eso se deben realizar una serie de pasos.

### **Paso 3.1: Análisis de las fuentes de datos**

La fuente de datos constituye un elemento fundamental ya que contiene información valiosa para la organización. Es necesario para el análisis de las fuentes de datos, haber tenido una reunión con los interesados, con el objeto de identificar las fuentes de datos existentes en la organización, seleccionar la(s) que se utilizará(n) para nutrir el *dashboard*, entender su(s) estructura(s) y cómo acceder a esos datos. Esto permitirá a los diseñadores del *dashboard* entender claramente los datos que se deben analizar.

En cualquier punto, un *dashboard* estará limitado por el tipo y calidad de los datos con los que se vaya a trabajar. Antes de crear contenido para un *dashboard*, debe completarse un análisis exhaustivo de los datos del entorno para descubrir objetos y métricas que se relacionan con el objetivo de negocio y profundizar en el impacto que tienen para la organización.

Se debe conocer a profundidad la fuente de datos para poder extraer los datos que se visualizarán posteriormente en el *dashboard*.

### **Paso 3.2: Selección de los datos**

Una vez que se analizaron las fuentes de datos, se puede comenzar con el proceso de selección de los datos, en este punto se recomienda escoger los que sean importantes e incidan en la toma de decisiones, datos que se desee realizar un seguimiento de su comportamiento en el tiempo, basándose en los datos históricos acumulados de la organización. Hay que tener en cuenta que no todos los datos almacenados en una fuente de datos poseen el potencial para ser utilizados, deben ser específicos, continuos y periódicos, objetivos, cuantificables, medibles, que propicien una alta calidad y esto se logra seleccionando datos que sean exactos, íntegros, coherentes, confiables y accesibles.

Los autores de esta investigación proponen que se discriminen los datos nulos, vacíos o duplicados y se utilice la herramienta *DataCleaner* en su versión 7.1 para obtener estadísticas e información sobre los datos de la fuente de datos seleccionada (los tipos de datos que predominan, el porcentaje por cada tipo de dato y la calidad de esos datos: valores nulos o vacíos del total de filas, cantidad de datos nulos por tabla, entre otros).

### **Paso 3.3: Estructuración de los datos**

Una vez se hayan seleccionado los datos que se deseen mostrar en el *dashboard* se requiere una forma particular de organizarlos con el objetivo de facilitar su manipulación. Deben ser enriquecidos para poder considerarse compatibles con el sistema que los va a consumir.

Se recomienda utilizar el formato de texto estándar JSON. Para llevar a cabo la estructuración de los datos en un JSON se deben tener en cuenta las siguientes características:

- JSON es un formato de datos que contiene sólo propiedades, no métodos.
- Se requiere usar comillas dobles para las cadenas y los nombres de propiedades, las comillas simples no son válidas.
- Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione.
- Los datos pueden tomar la forma de cualquier tipo de datos (cadenas, números, booleanos) que sea válido para ser incluido en un JSON, no solo arreglos u objetos.
- Sólo las cadenas entre comillas pueden ser utilizadas como propiedades (Friesen, 2019).

El gestor de base de datos *postgreSQL* tiene funciones implementadas para convertir datos a estructuras JSON.

Se muestra en la figura 2 un ejemplo de datos estructurados en el formato JSON, las propiedades se encuentran entre comillas, seguido de dos puntos y se asocia a cada propiedad un valor, los datos de tipo cadena se encuentran entre comillas y el resto son valores numéricos

### **Paso 3.4: Almacenamiento de los datos**

Una vez seleccionado y estructurado los datos que se quieren analizar, se deben almacenar en una base de datos, se exhorta se almacenen en bases de series de tiempo. Se recomienda almacenar los datos en InfluxDBv1.7.1.

```
{
  "nombre": "clara",
  "edad": 28,
  "profesor": true,
  "members": [
    {
      "departamento": "Ingenieria de software",
      "asignatura que imparte": "Ingenieria de software",
      "años de experiencia": 3,
    }
  ]
}
```

Figura 2. Ejemplo de datos estructurados en formato JSON

A continuación, se muestran operaciones básicas de InfluxDB:

- Comando para crear bases de datos en influxdb: create database “nombre de la base de datos”.
- Comando para mostrar las bases de datos almacenadas en influxdb: show databases.
- Comando para seleccionar la base de datos con la cual se trabajará: use “nombre de la base de datos”.
- Comando para mostrar las medidas que se tienen en la base de datos seleccionada: show measurement.
- Comando para mostrar toda la información de un indicador específico: select \* from “indicador”.
- Comando para mostrar usuarios: show users.
- Comando para crear usuarios: create user “username” with password ‘password’.
- Comando para crear usuarios administradores: create user “username” with password ‘password’ with all privileges.

#### FASE 4: CONSTRUCCIÓN DEL DASHBOARD

Se deben evaluar las herramientas de visualización de indicadores para el diseño de *dashboard*, los tipos de gráficos que mejor representan los datos que se van a mostrar. Se deben tomar decisiones sobre la presentación de la información para proporcionar mayor visibilidad al analizar esta. También se definen las alertas visuales, la configuración de umbrales y los cambios de color de los gráficos cuando los valores oscilen dentro de los umbrales definidos, así como configurar la actualización de los datos y la seguridad que estos tendrán. Seguidamente se explican cada uno de los pasos que están presentes en esta fase.

##### Paso 4.1: Seleccionar herramienta de visualización de indicadores

Se debe realizar una búsqueda exhaustiva de herramientas de visualización de indicadores que han sido desarrolladas y con las que se han diseñado *dashboard*, evaluar sus potencialidades y características técnicas, para luego seleccionar la herramienta que se ajuste a las necesidades existentes.

Se sugiere utilizar la herramienta *Grafana* en su versión 6.6.1 para el diseño de *dashboard*.

## Paso 4.2: Estudio de las principales funcionalidades de la herramienta de visualización seleccionada

Los autores de este artículo aconsejan consultar el sitio oficial de la herramienta de visualización de indicadores que se escogió en el paso anterior, y revisar la documentación técnica que explica detalladamente como trabajar con esa herramienta. A continuación, se muestran varios enlaces de documentación técnica y tutoriales de herramientas de visualización de indicadores:

- *Grafana* (documentación técnica): <https://grafana.com/docs>.
- Tutorial donde explican cómo diseñar *dashboard* en *Grafana* desde cero: <https://www.linuxito.com/cloud/1104-creando-mi-primer-dashboard-en-grafana>.
- *Kibana* (documentación técnica): <https://www.elastic.com/es/what-is/kibana>.
- Otras herramientas de visualización de indicadores (Tableau, sitio oficial): [www.tableau.com](http://www.tableau.com).

## Paso 4.3: Seleccionar tipos de gráficos que mejor se ajusten a los datos

Para seleccionar los tipos de gráficos que mostrará el *dashboard* se deben tener en cuenta los tipos de gráficos que existen y qué tipo de información se debe mostrar en cada uno, a continuación, se explican estos dos elementos (Barros, 2013):

- **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.
- **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.
- **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.
- **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.
- **Histograma:** se emplea para ilustrar muestras agrupadas en intervalos. Está formado por rectángulos unidos a otros, cuyos vértices de la base coinciden con los límites de los intervalos y el centro de cada intervalo es la marca de clase que representamos en el eje de las abscisas.
- **Gráfica gauge:** tiene forma de velocímetro de un automóvil, se utilizan para medir el desempeño actual y muestran ese valor utilizando una escala. El valor de cada aguja se lee fácilmente dentro del rango de datos coloreado o el eje del gráfico.
- **Cartograma:** las visualizaciones de mapas organizan los datos geográficamente, se utiliza para mostrar ubicaciones geográficas (Cámara, 2019). Un cartograma es un mapa en

el que se presentan datos estadísticos por regiones, bien poniendo el número o coloreando las distintas zonas en función del dato que representan.

- **Tablas:** las tablas organizan los datos en filas y columnas, se utilizan para mostrar conjuntos de datos bidimensionales que pueden ser organizados categóricamente (Cámara, 2019).

#### Paso 4.4: Presentación de la información

En el diseño de *dashboard*, se debe tener en cuenta cómo se realizará la presentación de la información para que el *dashboard* sea fácil de entender y tenga claridad en las métricas que se van a monitorear.

Según Stephen Few: “El contenido del tablero debe estar organizado de manera que refleje la naturaleza de la información y que admita un monitoreo eficiente y significativo. La información no puede colocarse en cualquier lugar del tablero, ni las secciones de la pantalla pueden dimensionarse simplemente para adaptarse al espacio disponible. Elementos que se relacionan entre sí, por lo general, deben colocarse cerca uno del otro. Los elementos importantes a menudo deben aparecer más grandes, por lo tanto, visualmente más destacados, que los elementos menos importantes. Los elementos que deben escanearse en un orden particular deben organizarse en secuencia para garantizar la atención visual” (Few, 2007). A continuación, se detallan elementos fundamentales a tener en cuenta para la presentación de la información:

- Definir nombres claros y consistentes: se deben nombrar y explicar claramente las métricas, unidades de medidas y valores que se muestran en el *dashboard* (Cámara, 2019).
- Colores: el uso de los colores, al representar los datos dentro de un *dashboard*, debe facilitar la visualización de la información; por tanto, se sugiere representar los valores negativos o disminuciones en color rojo y, por el contrario, los valores positivos o incrementos en color verde. Se propone utilizar una gama de colores simple y evitar colores brillantes (Mueses, 2020).
- Fuente: entre las fuentes más comunes se encuentran *Georgia*, *Times New Roman* y *Palatino*. *Georgia* es una fuente que funciona bien en la pantalla, en diferentes tamaños y en negrita y cursiva. *Palatino* es elegante, mientras que *Times New Roman* es profesional. Otras de las fuentes que se recomienda utilizar son *Arial*, *Helvética*, *Trebuchet*, *Verdana* y *Tahoma*. La fuente que se escoja para mostrar la información es elección de la persona que diseñe el *dashboard* (Juice, 2009).
- El uso de paneles: un *dashboard* no debe mostrar demasiada información (Rojas, 2019), por tanto, se aconseja dividir en paneles la interfaz del *dashboard*, agrupando los indicadores que se encuentren relacionados.
- Posición de la información importante: la información relevante (métricas y KPI vistos con mayor frecuencia) se deben posicionar en la esquina superior izquierda del *dashboard* (Rojas, 2019) ya que los estudios muestran que la mayoría de las personas tienden a escanear una página de manera similar. La investigación indica que los usuarios buscan primero información en la parte superior e izquierda. El centro también recibe bastante atención. Pero es posible que los usuarios se fijen muy poco en la parte inferior y la derecha de la interfaz (Juice, 2009).

#### Paso 4.5: Seleccionar funcionalidades básicas y avanzadas

En las herramientas de visualización luego de escoger el tipo de gráfico se pueden configurar varios elementos interactivos que resalten la información clave en un *dashboard*. Seguidamente, se explican varias de las funcionalidades básicas y avanzadas que pueden estar presente en un *dashboard* (Juice, 2009).

##### Funcionalidades básicas:

- **Vista resumen-detalle:** es la capacidad de pasar de una métrica o vista de resumen a detalles adicionales que brindan más contexto e información sobre las métricas.
- **Filtros:** permiten a los usuarios definir el alcance de los datos en el *dashboard* para reflejar sus necesidades. Los filtros pueden ser globales (alcance para todo el *dashboard*) o locales (alcance para un gráfico, métrica o vista específica).
- **Comparación de datos:** capacidad de mostrar dos o más subconjuntos de datos uno al lado del otro o sobre el mismo gráfico.
- **Exportar:** brinda a los usuarios la capacidad de extraer información de un *dashboard* para ser utilizada y visualizada en otros formatos (excel, pdf, csv).
- **Umbrales:** son un valor límite para la presencia o no de una determinada condición en la información del gráfico, se compone de dos partes: la condición y el formato que se aplicará a los valores que cumplan las condiciones. La complejidad de las condiciones puede variar dependiendo del tipo de umbral creado. Se considera que los datos cumplen la condición cuando han pasado el umbral de dicha condición (Grafana, 2020). Los umbrales son de gran utilidad en la representación de información en gráficos, ya que permiten resaltar ideas específicas o identificar valores atípicos, así como definir colores para diferenciar la información.
- **Alertas:** son un mecanismo para enfocar las excepciones, los valores atípicos y los datos destacados, pueden ser la capa adicional de abstracción que hace que un *dashboard* sea útil (Barros, 2013). Resaltan información basada en criterios predefinidos, las alertas pueden activarse cuando una métrica sale de un umbral particular y permiten mostrar ciertas condiciones de los distintos recursos monitoreados, llevando un control del comportamiento y el estado en que se encuentra la información en el gráfico.
- **Notificaciones de alertas:** se utilizan para comunicar a los usuarios cuando una alerta cambia el estado en el que se encuentra, enviando un mensaje por diferentes vías (correo electrónico, red social *Telegram*, *Google Hangouts Chat*, entre otras).

##### Funcionalidades avanzadas:

- **Resumen basado en texto:** es una descripción textual generada automáticamente de la información clave en el *dashboard*.
- **Etiquetado:** permite identificar información importante para los usuarios en una palabra o frase corta.
- **Anotación:** permite a los usuarios agregar comentarios a números o gráficos específicos.
- **Guardar cambios:** permite a los usuarios guardar las configuraciones que han realizado en el *dashboard*.

- Visualizaciones avanzadas: se utiliza para mostrar datos complejos en el *dashboard* por medio de mapas geográficos, diagrama de dispersión, entre otros.

Los autores de esta investigación consideran que los *dashboard* que se diseñen utilizando esta metodología no tienen que cumplir con todas las funcionalidades básicas y avanzadas mencionadas anteriormente, sino que los diseñadores tienen la libertad de seleccionar cuáles de ellas incluirán en el *dashboard*.

#### Paso 4.6: Actualización y seguridad de los datos

En los *dashboard*, para garantizar la seguridad y privacidad de la información se deben gestionar usuarios para las personas que interactuarán con el sistema y que accedan a la base de datos, se sugiere que se definan roles para agrupar a los diferentes usuarios. La herramienta Grafana contiene su propio protocolo de seguridad proporcionando privacidad de la información a través de la gestión de usuarios y roles.

Se recomienda acceder a la aplicación mediante un protocolo de seguridad, se sugiere el protocolo HTTPS, pues este proporciona confidencialidad e integridad de los datos. Se deben crear además políticas de respaldo y recuperación de información para garantizar la integridad de los datos, de manera que se encuentren almacenados en un lugar seguro en caso de fallos en el sistema, por tanto, se propone que se realicen salvadas periódicas y copias de seguridad frecuentemente. Debe definirse también la periodicidad (semanal, quincenal, mensual, etc.) con que actualizará la información almacenada en el *dashboard*.

### FASE 5: VALIDACIÓN DEL DASHBOARD

En esta fase se propone una estrategia de pruebas para demostrar que el sistema cumple con los requisitos funcionales y no funcionales definidos.

Las estrategias de pruebas de *software* proporcionan una guía que describe los pasos que deben realizarse como parte de las pruebas. Estas deben incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (Pressman, 2005).

En la siguiente tabla se muestra la estrategia de pruebas diseñada, que puede ser aplicada con el objetivo de comprobar el funcionamiento del *dashboard* creado.

Tabla 3. Estrategia de Pruebas. Fuente: Elaboración propia.

Nivel de Prueba	Tipo de Prueba	Método	Herramientas	Roles	Objetivos de los niveles
Sistema	Funcional	Caja negra	Casos de prueba	Probadores	-Validar que el sistema está completo y funciona como se espera. -Verificar que los requisitos funcionales y no funcionales del sistema son los especificados (Sánchez, 2019).
	Rendimiento		Apache JMeter		
	Seguridad		Acunetix Web Vulnerability Scanner		
	Portabilidad		Navegadores web		
Aceptación	Alfa	Caja negra	-Casos de Prueba -Manual de usuario	Analistas y usuarios finales	
	Usabilidad		-Lista de chequeo		



### VALIDACIÓN DE LA METODOLOGÍA PROPUESTA. TÉCNICA IADOV.

Para valorar y/o constatar el nivel de aceptación y la aplicabilidad de la propuesta de solución se utilizó la técnica de Iadov. Para el desarrollo de esta técnica se encuestó un grupo de siete especialistas de diferentes proyectos del centro CREAD, donde la composición de los roles varía entre analistas, administradores de bases de datos, desarrolladores y jefes de proyecto, teniendo como promedio cinco años de experiencia en el rol.

Mediante esta técnica se puede determinar, de forma indirecta, Índice de Satisfacción Grupal (ISG) de los individuos involucrados en el proceso que está siendo objeto de análisis, la misma ha sido aplicada en múltiples campos, como parte de las validaciones (Cañizares, 2012). Para la aplicación de la técnica Iadov se define un cuestionario conformado por cinco preguntas, de las cuales tres son cerradas y dos abiertas. Según (Ramírez, 2018), las preguntas cerradas guardan relación entre sí, esto no es de conocimiento por parte del sujeto al que se le aplica la técnica.

Las tres preguntas se relacionan a través del denominado “Cuadro Lógico de Iadov” el cual permite ubicar a cada encuestado en una escala de satisfacción, para luego calcular el ISG. Las respuestas a cada una de estas preguntas permiten determinar la posición de cada sujeto en la escala de satisfacción que toma valores desde 1 hasta 5, distribuido de la siguiente manera: 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definido o contradictorio, 4-Más insatisfecho que satisfecho y 5-Clara insatisfacción (Ramírez, 2018). El cuadro lógico utilizado en la investigación se muestra en la siguiente tabla.

Tabla 4. Cuadro lógico de Iadov. Fuente: Elaboración propia.

¿Cuál es su criterio sobre la metodología para el diseño de dashboard para los proyectos del centro CREAD?	¿Desarrollaría usted un dashboard sin utilizar una metodología para su diseño?								
	Sí			No sé			No		
	¿Utilizaría usted la metodología propuesta para diseñar un dashboard para su proyecto?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Clara satisfacción	1	2	3	2	2	3	3	3	2
Más satisfecho que insatisfecho	2	2	3	2	3	3	2	3	2
No definido o contradictorio	3	3	3	3	3	3	3	3	3
Más insatisfecho que satisfecho	2	3	3	3	4	4	3	3	4
Clara insatisfacción	2	3	3	3	4	4	3	3	5

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1)+B(+0.5)+C(0)+D(-0.5)+E(-1)}{N}$$

Las variables representan las cantidades de participantes agrupados por las escalas del índice de satisfacción individual. Donde:

- **A:** representa la cantidad de participantes que expresaron tener una clara satisfacción.
- **B:** representa la cantidad de participantes que se sienten más satisfechos que insatisfechos.
- **C:** representa la cantidad de participantes que expresaron contradicción o no definido.
- **D:** representa la cantidad de participantes que se sienten más insatisfechos que satisfechos.
- **E:** representa la cantidad de participantes que expresan una clara insatisfacción.
- **N:** representa el total de participantes encuestados.

Para obtener el índice de satisfacción grupal (ISG) se procesan los criterios de las personas de acuerdo a los niveles de satisfacción que se expresan en la escala numérica que oscila entre -1 y +1. Es por ello que, una vez calculado, los valores que se encuentren comprendidos entre -1y -0,5 indican insatisfacción; los comprendidos entre -0,49 y + 0,49 evidencian contradicción y los que se ubiquen entre 0,5 y 1 indican que existe satisfacción. Al calcular el ISG queda de la siguiente manera:

$$ISG = \frac{A(+1)+B(+0.5)}{N}$$

$$ISG = \frac{(4(+1)+3(+0.5))}{7}$$

$$ISG = 0.79$$

En este caso el valor del ISG fue de 0,79 lo que indica satisfacción con respecto a la metodología propuesta. La técnica de Iadov contempla además preguntas complementarias de carácter abierto de las cuales se obtuvieron valoraciones positivas.

El proceso de validación de la propuesta de solución mediante la aplicación de una encuesta a especialistas del centro CREAD y el análisis de los resultados obtenidos mediante la técnica de

Iadov confirmó su posibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG= 0.79). Además, los criterios emitidos en las preguntas abiertas reflejan la aceptación de la propuesta y un reconocimiento a su utilidad en función de su aporte práctico.

### **Dashboard desarrollado aplicando la metodología propuesta**

Seguidamente se muestran imágenes del *dashboard* diseñado para el proyecto SIGEv3.0 del Ministerio de Educación Superior (ver figuras 3 y 4). Se comprobó que el *dashboard* satisface las necesidades del cliente posibilitando visualizar el comportamiento de indicadores seleccionados a través de gráficos, tablas resúmenes, y paneles. Para el desarrollo del *dashboard* se tuvieron en cuenta las fases y pasos explicadas en la metodología propuesta.

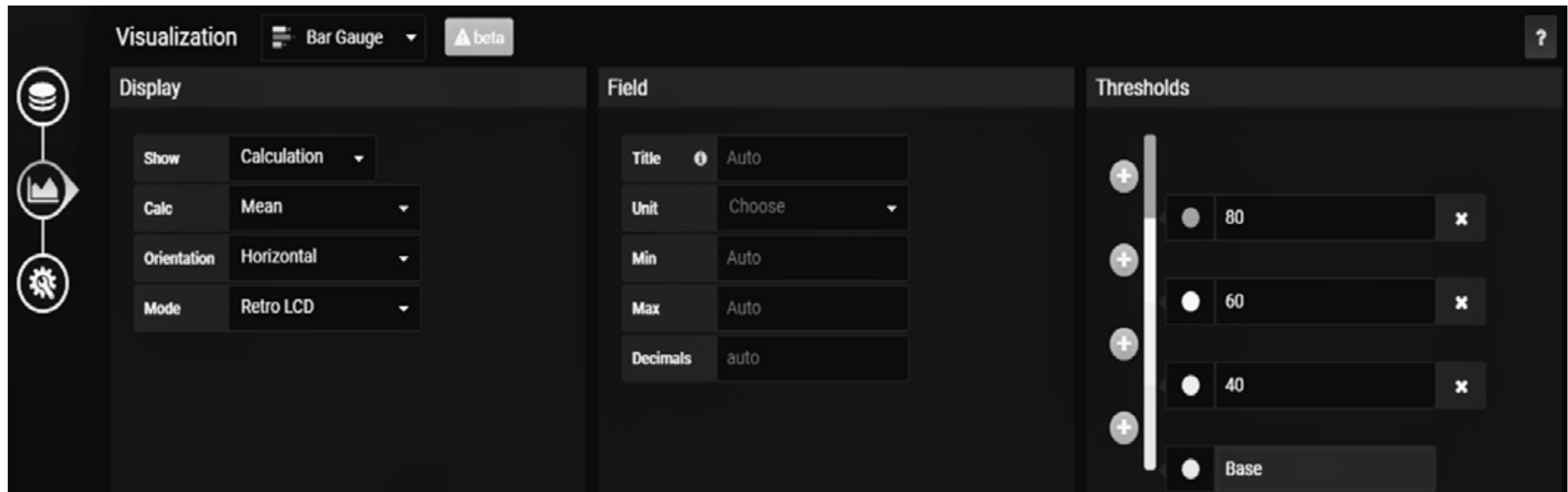


Figura 3. Configuración de propiedades y umbrales de un gráfico de barra.

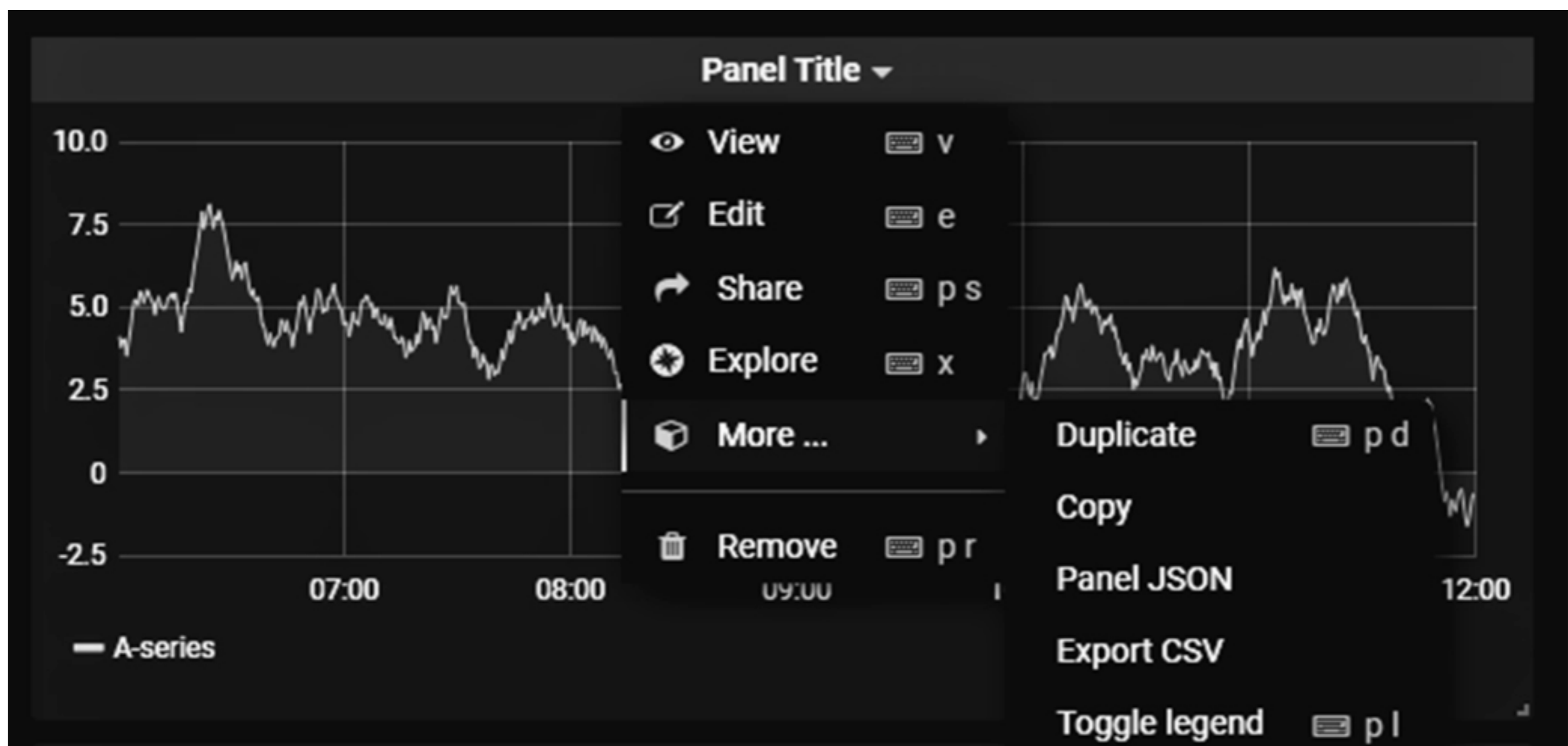


Figura 4. Exportar los datos de un gráfico a formato CSV.

## CONCLUSIONES

A partir del estudio del arte de las principales metodologías para el desarrollo de *dashboard* se pudo constatar diferentes puntos de vista de los autores para llevar a cabo la implementación exitosa de un *dashboard* en una organización, de tal manera que cada una de ellas aportó especificaciones que fueron tomadas en cuenta para la realización de la metodología propuesta. Como resultado de ese estudio, se decidió tomar como base para la creación de la metodología propuesta la Metodología de *Noetix Corporation*, en función de garantizar un adecuado entendimiento se incluyeron pasos específicos dentro de cada fase que orienten a los desarrolladores.

Las metodologías estudiadas incluyen de forma superficial las pruebas de *software* (no recomiendan niveles y tipos de pruebas a realizar), además obvian la seguridad de los datos. En la metodología propuesta la seguridad de los datos se incluyó como un paso dentro la fase cuatro “Construcción del *dashboard*” y en la fase cinco “Validación del *dashboard*” se detalló la estrategia de pruebas a seguir.

La validación científica a través de la técnica de Iadov, evidenció el nivel de aceptación de la propuesta por parte de los especialistas encuestados.

La aplicación de la metodología en el diseño de un *dashboard* para el proyecto SIGEv3.0 del MES permitió confirmar de forma positiva la hipótesis planteada en la investigación

## REFERENCIAS

- Ballesteros de Andrés, C., & Sypko, D. (2017). Sistema IoT para monitorización del estado de un centro de proceso de datos de grandes dimensiones. Trabajo de Fin de Grado en Ingeniería de Computadores. Universidad Complutense de Madrid. Obtenido de <https://bit.ly/3C6YtP2>
- Barros, R. M. P. D. C. (2013). *Dashboarding: projeto e implementação de painéis analíticos*. Tesis de Maestría. Universidad de Minho. Obtenido de <https://bit.ly/3k3tH3m>
- Cañizares, R. (2012). Repositorio de recursos educativos para las instituciones de educación superior. Tesis de Doctorado. Universidad de las Ciencias Informáticas, La Habana, Cuba.
- DB-Engines. (2020a). DB-Engines Ranking of Search Engines. Obtenido de <https://bit.ly/3lcGpfE>
- DB-Engines. (2020b). DB-Engines Ranking of Time Series DBMS. Obtenido de <https://bit.ly/3EeROV8>
- DB-Engines. (2020c). Elasticsearch System Properties. Obtenido de <https://bit.ly/3ligl2H>
- DB-Engines. (2020d). InfluxDB System Properties. Obtenido de <https://bit.ly/2XcOWqs>
- DB-Engines. (2020e). System Properties Comparison Elasticsearch vs. InfluxDB. Obtenido de <https://bit.ly/3Ed8ImY>
- Elasticsearch. (2020). *Documentación técnica de Elasticsearch*. Obtenido de <https://bit.ly/3z6lgc7>
- Edis, R. (2016). *Using Agile Methods with BI Dashboard Development*.
- Few, S. (2007). *Pervasive Hurdles to Effective Dashboard Design, Visual Business Intelligence Newsletter*.
- Ferrer, D. P., Betancourt, A. G., Fernández, R. E. M., & Beltrán, L. A. G. (2019). Arquitectura DATA WAREHOUSING para la operación de los Servicios de Tecnologías de la Información de ETECSA. Tono, Revista Técnica de la Empresa de Telecomunicaciones de Cuba SA, 15(1), 66-76.
- Grafana, Labs. (2020). Documentación técnica de Grafana. Obtenido de <https://bit.ly/3k6tb4Y>
- Influxdb. (2020). *Documentación técnica de InfluxDB*. Obtenido de <https://bit.ly/3A6coEH>
- Juice Inc. (2009). *A guide to creating dashboards people love to use*.
- Jusko, M. (2017). *A Methodology for Authoring Dashboards*.
- Martínez-Robalino, D. A. (2017). Metodología para el diseño de Dashboards orientado hacia el registro de evidencias en el proceso de evaluaciones institucionales. Tesis de maestría.
- Mueses, R. A. P. (2020). Implementación de un *dashboard* de alertas. Sistema, 10, 3. Trabajo de grado presentado previo a la obtención del título de Ingeniero en Sistemas Computacionales. Universidad Técnica del Norte. Obtenido de <https://bit.ly/3tA6mcQ>

- Morales Padron, M. (2016). Socialize. Sistema de sincronización dinámica de interfaces de usuario acoplables a servicios de minería de datos. Tesis de pregrado. Universidad de Alicante. Obtenido de <https://bit.ly/3tAKD4m>
- Nazco Torres, W. (2018). Almacenamiento y visualización de series temporales. Trabajo de fin de grado en Ingeniería Informática. Universidad de La laguna. Obtenido de <https://bit.ly/2X9zHPa>
- Orts, D. (2004). Dashboard development and deployment. Bellevue: Noetix Corporation.
- Perez-Montoro, M. (2017). Visualización de información en 2016: conceptos, contenidos y tecnología. Anuario ThinkEpi, Vol. 11, pags. 284-296. ISSN 2564-8837
- Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave macmillan.
- Quesada, A. Y., Wong, D., & Rosete, A. (2008). Minería de Datos aplicada a la Gestión Hospitalaria. 14 Convención Científica de Ingeniería y Arquitectura, 2-5.
- Ramírez, Marlon. (2018). Aplicación de la técnica de Iadov.
- Rojas Ortiz, A. (2019). Diseño de un Cuadro de Mando Integral para gestión ágil de proyectos de desarrollo de *software*. Máster Universitario en Ingeniería y Tecnología de Sistemas Software. Unversidad Politécnica de Valencia. Obtenido de <https://bit.ly/2Xgt9i0>
- Sainz Gilarranz, Rubén (2020). Automatización de infraestructura para orquestación, gestión y monitorización de aplicaciones basadas en contenedores en el dominio IoT Edge y despliegue continuo de *software*. Tesis de maestría en Software de Sistemas Distribuidos y Empotrados. Universidad Politécnica de Madrid. Obtenido de <https://bit.ly/2Xb97pp>
- Sánchez, G. A. R. (2019). Metodología ágil orientada al desarrollo de proyectos de *software* en trabajos de grado de ingeniería. Encuentro Internacional de Educación en Ingeniería.
- Soler González, R. H. (2009). Procedimiento para la implementación del Balanced Scorecard como modelo de gestión en las empresas cubanas. Tesis para optar por el grado científico de Doctor en Ciencias Técnicas. CUJAE.
- Soler González, R. H. (2009). Cuadro de mando (2009-09). (S. A. SL, Ed.) Revista Contribuciones a la Economía. Obtenido de <https://bit.ly/3tAfEWc>. ISSN 1696-8360.

Copyright © 2021 Cordova-Viera, Y., Martinez-Borrego, J., Cordova-Viera, E.



Este obra está bajo una licencia de Creative Commons Atribución-No Comercial 4.0 Internacional