

# Visualización de datos de GrimoireLab con Grafana

Jonathan Cano Picazo

Abril 2020

## 1 Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

## 2 Desarrollo e implementación

- Metodología SCRUM
- Iteraciones o sprints

## 3 Diseño y resultados

- Flujo lógico del proyecto
- Evaluación y demostración de la herramienta
- Extensiones o plugins
- Comparativa de herramientas

## 4 Conclusiones

- Consecución de objetivos
- Conocimientos aplicados
- Lecciones aprendidas
- Trabajos futuros

## 5 Referencias

# Descripción del problema

¿Qué es un dashboard?

**Dashboard:** Representación gráfica de métricas y datos clave.



Figure: Ejemplo de cuadro de mandos

# Descripción del problema

¿Qué herramientas nos permiten representarlos?



Grafana, Kibana, Klipfolio y Google Data Studio, respectivamente.

# Descripción del problema

La herramienta elegida

**Grafana**, el software elegido para este proyecto:

- Herramienta top dentro del sector
- Código abierto
- Énfasis en series temporales en BBDD
- Compatibilidad con infinidad de BBDD

# Descripción del problema

## Grafana

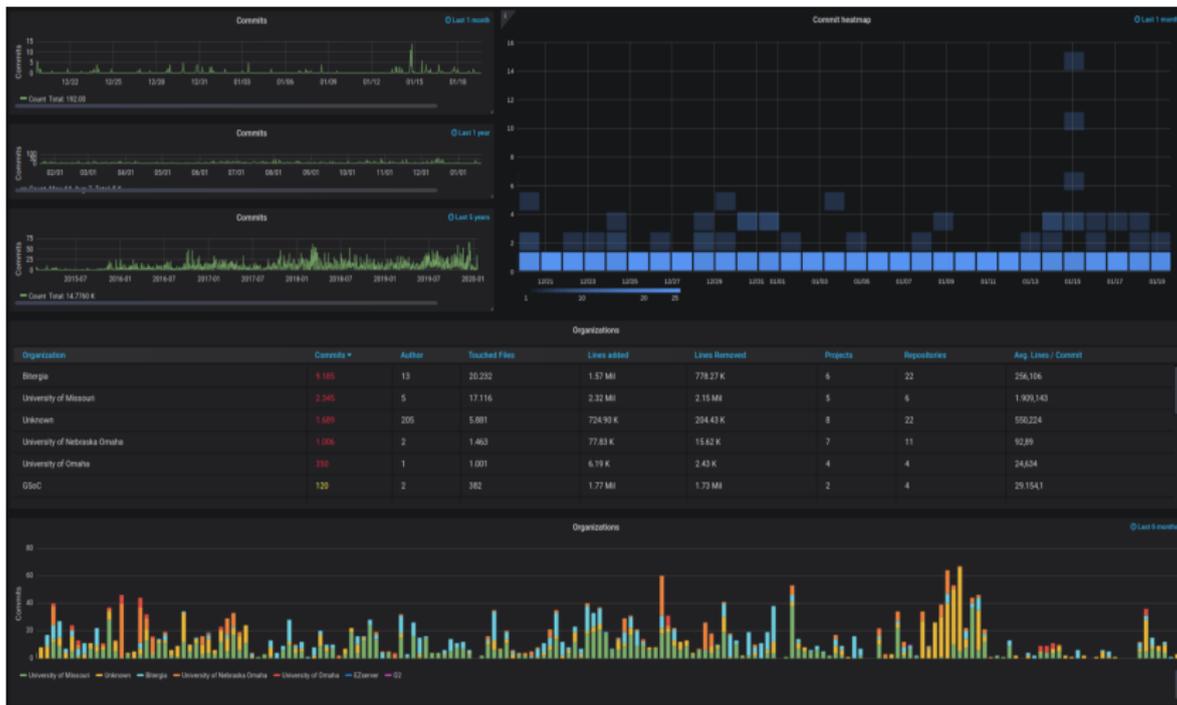


Figure: Ejemplo de dashboard usando Grafana

# Descripción del problema

¿Cómo obtenemos los datos? - GrimoireLab

- **Recopilar** datos de múltiples proyectos
- **Enriquecimiento** de los mismos
- **Visualización** de datos con Kibana



# Descripción del problema

## GrimoireLab y Kibana

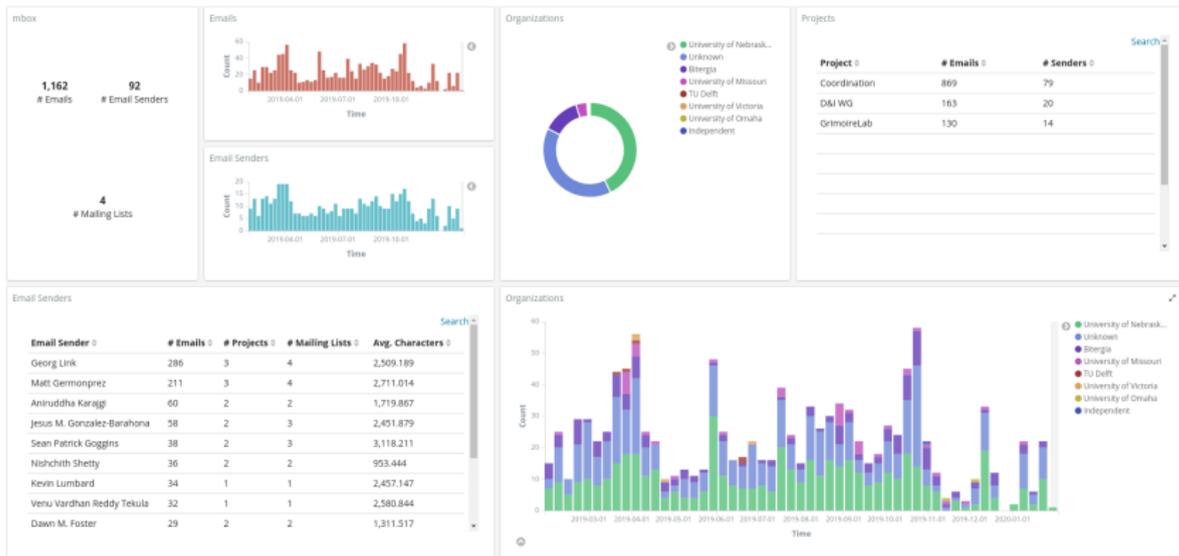


Figure: Dashboard en Kibana, usando los datos de GrimoireLab

# Objetivos principales

- **Representación** de datos de GrimoireLab con Grafana, frente al uso de Kibana usado hasta ahora.
- **Comparación** de las dos herramientas de representación utilizadas: Grafana y Kibana. Para ello se exploran los límites de cada una de ellas (**plugins** p.e.), señalando además sus semejanzas y diferencias.
- Preparación de un prototipo que permita **unificar** todas las herramientas en una misma imagen de contenedor, usando la tecnología *Docker*.
- Uso de **multicontenedores**<sup>1</sup>: despliegue de varios contenedores de distintos proyectos<sup>2</sup>, ahorrando tiempo en la extracción, enriquecimiento y visualización de datos.

---

<sup>1</sup><https://www.youtube.com/watch?v=7He7GWpa64s>

<sup>2</sup><https://www.youtube.com/watch?v=a-uYU0tXhw0>

# Objetivos principales

Integración / Unificación

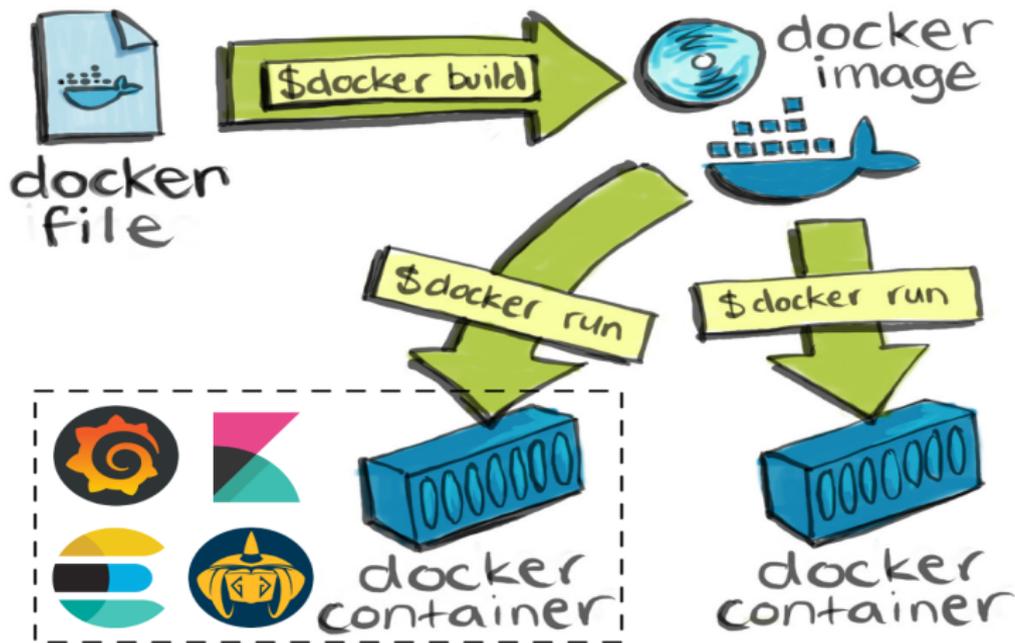


Figure: Creación y despliegue de una imagen y contenedor mediante Docker

- 1 Bases de datos
  - Elasticsearch, MariaDB, InfluxDB.
- 2 Sistemas de visualización
  - Grafana
  - Kibana
  - Cauldron
- 3 Otras tecnologías y servicios
  - Git / GitHub
  - GrimoireLab
  - Docker Hub
  - Overleaf

## 1 Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

## 2 Desarrollo e implementación

- Metodología SCRUM
- Iteraciones o sprints

## 3 Diseño y resultados

- Flujo lógico del proyecto
- Evaluación y demostración de la herramienta
- Extensiones o plugins
- Comparativa de herramientas

## 4 Conclusiones

- Consecución de objetivos
- Conocimientos aplicados
- Lecciones aprendidas
- Trabajos futuros

## 5 Referencias

# Desarrollo e implementación

## Metodología SCRUM

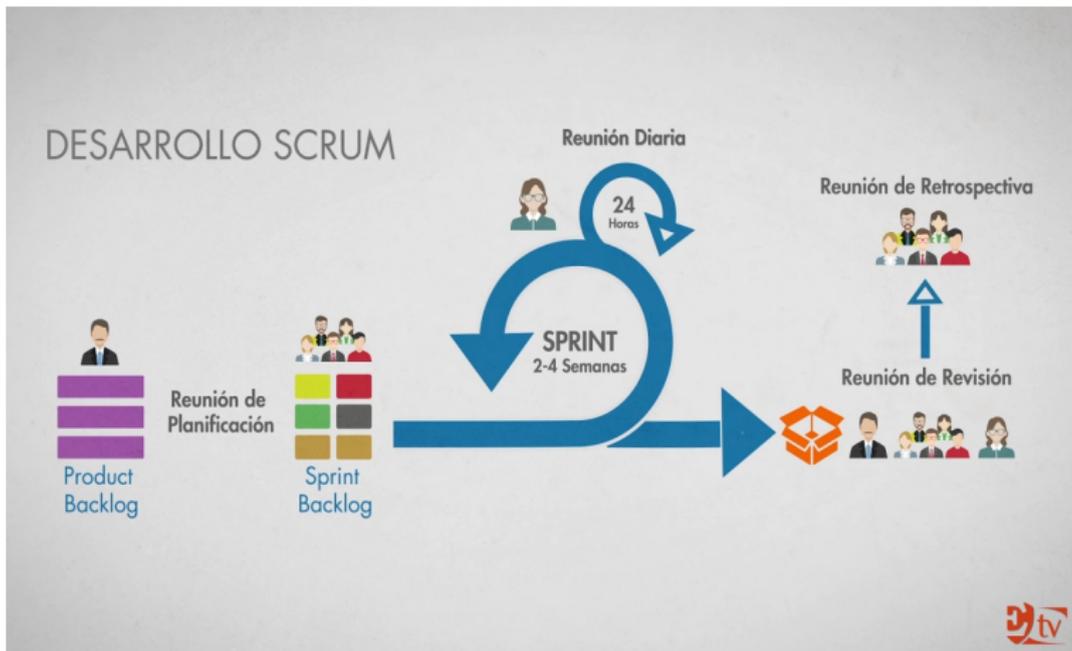


Figure: Esquema del entorno de trabajo SCRUM

# Desarrollo e implementación

## Iteraciones o sprints

- **Sprint 1** - Instalación de Docker Hub y GrimoireLab.
- **Sprint 2** - Grafana: instalación y utilización.
- **Sprint 3** - Primer panel y plugins de Grafana.
- **Sprint 4** - Dashboards funcionales y comparación.
- **Sprint 5** - Integración: construcción del contenedor de GrimoireLab/Grafana.
- **Sprint 6** - Actualización del contenedor de Grafana.
- **Sprint 7** - Imagen final: alojamiento en la nube y contenedores múltiples.

## 1 Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

## 2 Desarrollo e implementación

- Metodología SCRUM
- Iteraciones o sprints

## 3 **Diseño y resultados**

- Flujo lógico del proyecto
- Evaluación y demostración de la herramienta
- Extensiones o plugins
- Comparativa de herramientas

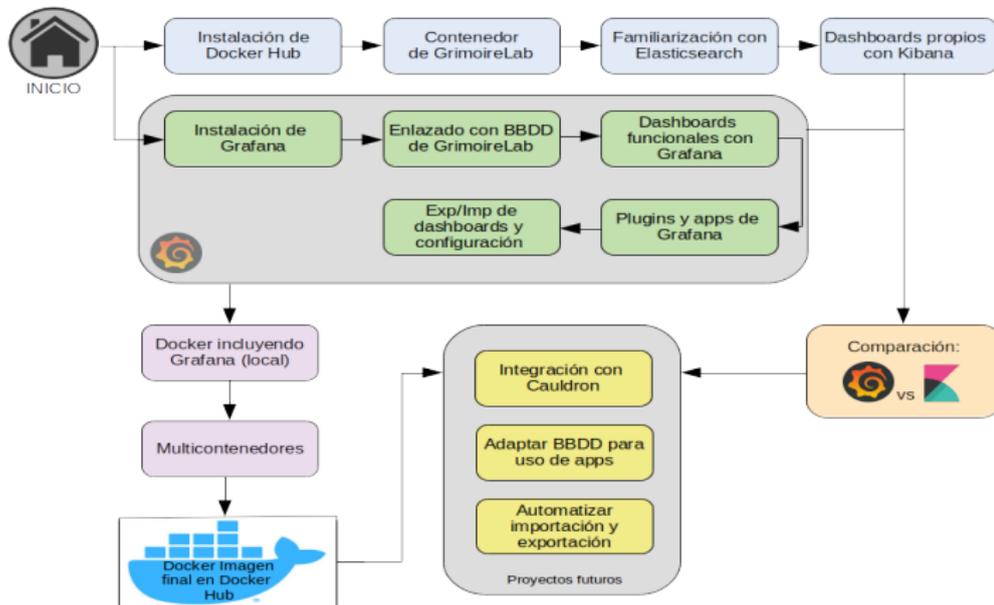
## 4 Conclusiones

- Consecución de objetivos
- Conocimientos aplicados
- Lecciones aprendidas
- Trabajos futuros

## 5 Referencias

# Diseño y resultados

## Flujo lógico del proyecto



# Diseño y resultados

## Demostración de la herramienta

- Git, Mailing Lists, Plugins, Pruebas.

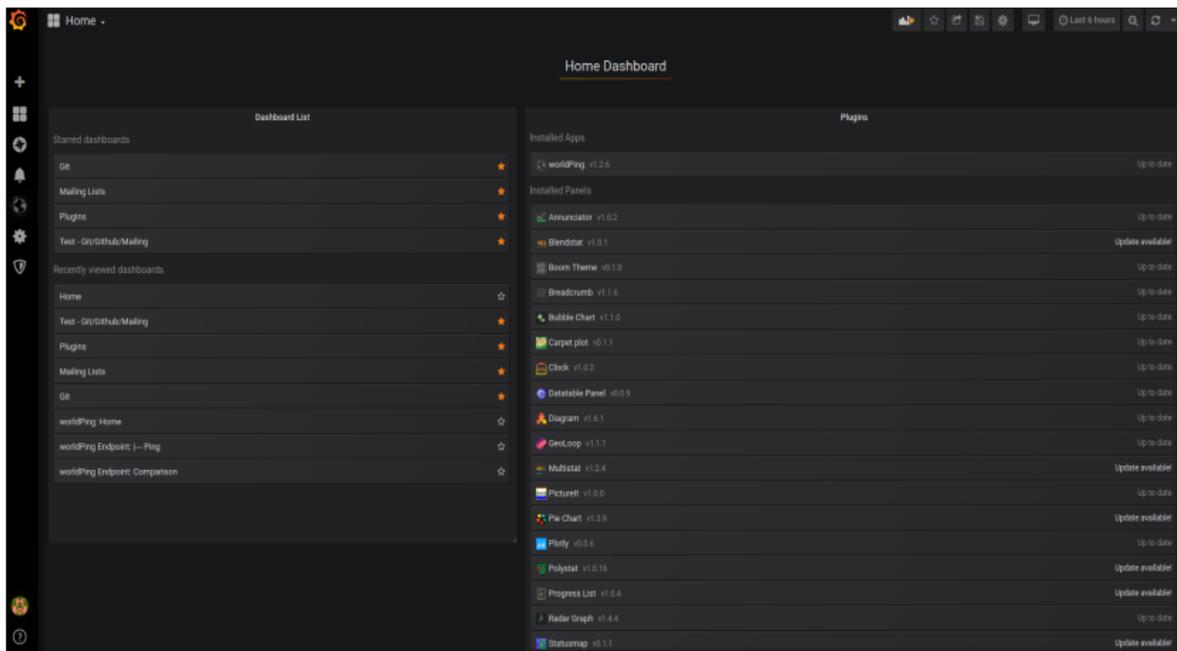


Figure: Dashboard *Home* de Grafana

# Diseño y resultados

## Demostración de la herramienta (2)

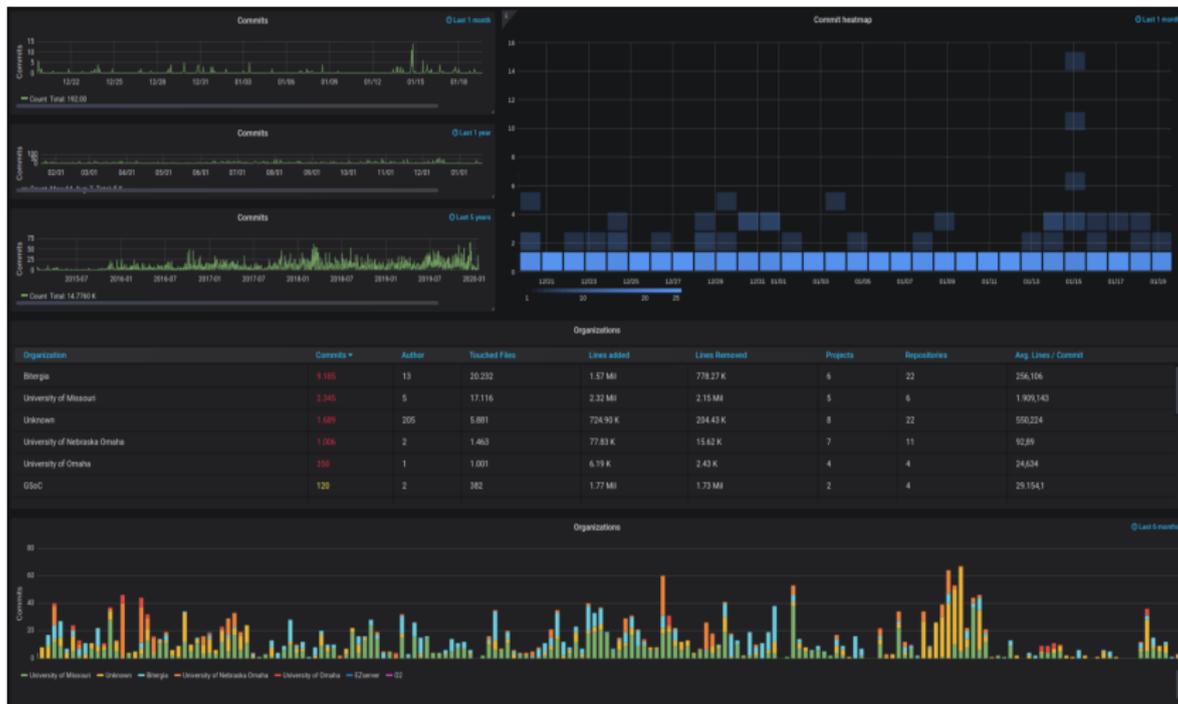


Figure: Extracto del dashboard *Git* realizado con Grafana

# Diseño y resultados

## Extensiones o plugins

**Plugin:** extensión para Grafana, a modo de nuevos paneles, nuevas fuentes de datos (DB) o apps.



Figure: Ejemplo de plugins (1)

# Diseño y resultados

## Extensiones o plugins (2)

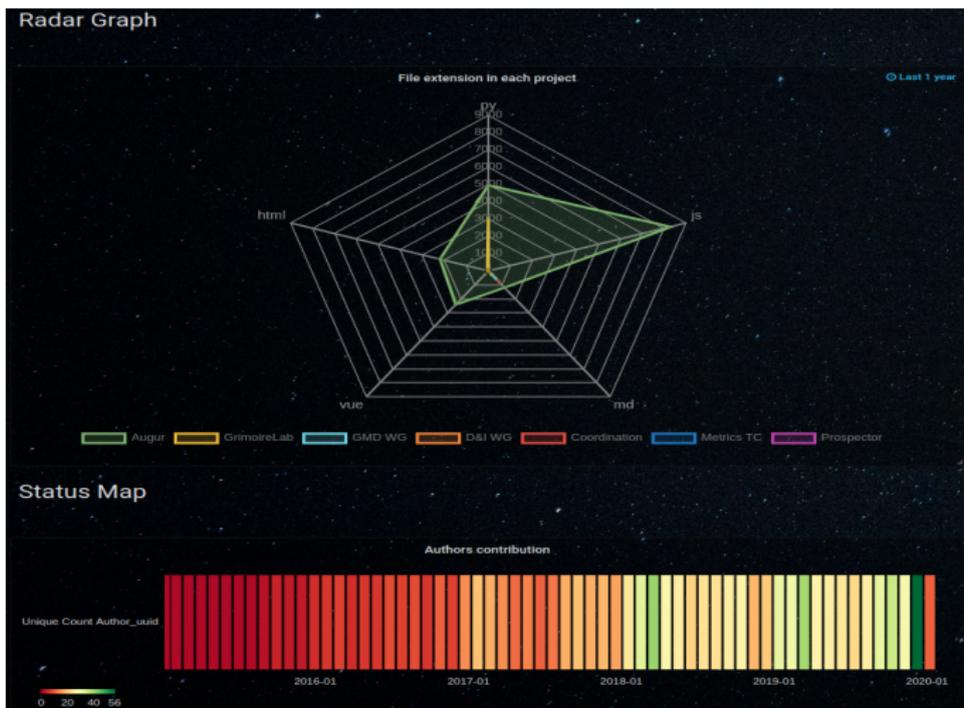


Figure: Ejemplo de plugins (2)

# Diseño y resultados

## Extensiones o plugins (3)



Figure: Ejemplo de plugins (3)

- Ejes de paneles.
- Manejo de documentos tipo *string*.
- Intervalos de tiempo.
- Selección de rangos temporales (gráficos).
- Bases de datos de los dashboards.
- Aspecto visual.
- Agrupación de paneles.
- Playlists.
- Plugins.
- Alertas.
- Exportación e importación.

## 1 Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

## 2 Desarrollo e implementación

- Metodología SCRUM
- Iteraciones o sprints

## 3 Diseño y resultados

- Flujo lógico del proyecto
- Evaluación y demostración de la herramienta
- Extensiones o plugins
- Comparativa de herramientas

## 4 Conclusiones

- Consecución de objetivos
- Conocimientos aplicados
- Lecciones aprendidas
- Trabajos futuros

## 5 Referencias

- **Representación** de datos con Grafana. Un total de 3 dashboards completos y completamente funcionales. ✓
- Explorar los límites de la herramienta (apartado de **plugins**). Se han usado un total de 20 plugins externos. ✓
- **Comparación**, mencionando numerosas ventajas/desventajas y similitudes/diferencias entre las dos herramientas (Grafana vs Kibana). ✓
- **Unificación** de las herramientas mediante el uso de Docker images<sup>3</sup>. ✓
- Creación de contenedores adicionales para el despliegue de **múltiples proyectos**. ✓

---

<sup>3</sup><https://hub.docker.com/r/onac8/grafana-grimoirelab>

# Conclusiones

Conocimientos aplicados (asignaturas asociadas)

- 1 Sistemas Operativos.
- 2 Servicios y Aplicaciones Telemáticas (SAT).
- 3 Ingeniería de Sistemas de Información (ISI).
- 4 Desarrollo de Aplicaciones Telemáticas (DAT).

# Conclusiones

## Lecciones aprendidas

- Uso de sistemas de visualización de datos (Grafana y Kibana).
- Análisis, recopilación, estudio y representación de datos de múltiples formas (aspecto más visual, orientarlo a un público concreto, etc.)
- Manipulación de bases de datos (queries, inserción, reestructuración...) de tipo NoSQL (Elasticsearch).
- Importación/exportación de BBDD en Grafana, haciendo uso de su API.
- Creación y manipulación de contenedores de imágenes completamente funcionales, gracias a Docker Hub.
- Integración de aplicaciones: permitir que dos o más herramientas trabajen conjuntamente, comunicándose sin fallos.

# Conclusiones

## Lecciones aprendidas (2)

- Permitir el despliegue web de este conjunto de herramientas integradas, gracias nuevamente al servicio Docker Hub.
- Posibilidad de trabajar en un proyecto de un tamaño considerable (CHAOSS).
- Perspectiva de trabajo en grupo: comunicación con mi tutor, feedback mutuo, discusión de ideas futuras...
- Organización y estructuración temporal de un proyecto de gran envergadura.
- Aprendizaje de LaTeX.

- Desplegar esta herramienta en la nube, en algún servidor de internet. Posibilidad de **integrar** esta aplicación en un proyecto de mayor envergadura (**Cauldron**<sup>4</sup>).
- Modificar los **índices** enriquecidos de la base de datos para que cuenten con documentos que hagan funcionar las diferentes Apps de Grafana (Worldmap, Worldping...).
- Creación de un sistema que nos permita **exportar/importar** las diferentes configuraciones de Grafana (usuarios, dashboards...) de una manera sencilla y automática. Para ello se puede hacer uso de la API de la herramienta y crear diferentes scripts dependiendo de lo que se quiera exportar/importar en un nuevo contenedor o localmente.

---

<sup>4</sup><https://gitlab.com/cauldronio/cauldron-deployment> 

- 1 Grafana Documentation.  
<https://grafana.com/docs/grafana/latest/>
- 2 Kibana Guide.  
<https://www.elastic.co/guide/en/kibana/current/index.html>
- 3 Elasticsearch Reference.  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- 4 SCRUM (software development methodology).  
[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))
- 5 GrimoireLab Tutorial.  
<https://chaoss.github.io/grimoirelab-tutorial/>
- 6 CHAOSS Project.  
<https://chaoss.community/>
- 7 Docker Hub Documentation.  
<https://docs.docker.com/docker-hub/>

- Web del proyecto.

`https://onac8.github.io/TFG-Grafana-Dashboard-Representation/`

- *Docker image* del proyecto.

`https://hub.docker.com/r/onac8/grafana-grimoirelab`

- Demo del proyecto.

`https://www.youtube.com/playlist?list=`

`PLCVLuotTGE7hU7RdRduHZcCPVar1wbXdD`