

Representación de datos de GrimoireLab con Grafana

Jonathan Cano Picazo

Marzo 2020

Contenido

① Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

② Desarrollo e implementación

- Metodología Scrum
- Iteraciones o sprints

③ Diseño y resultados

- Arquitectura final
- 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?

Un **dashboard** o cuadro de mando es una herramienta que permite representar, de manera visual, las diferentes métricas o datos de interés asociados a un proyecto concreto. Con los dashboard podemos analizar esta información y podemos observar cómo evoluciona un proyecto en el tiempo, detectar los posibles problemas de este, mantenimiento del mismo, etc.

Descripción del problema

¿Qué herramientas nos permiten representarlos?

Existen múltiples herramientas que nos permiten representar dashboards. Entre ellas se encuentran: Kibana, Google Data Studio, Klipfolio, **Grafana**, etc. Este proyecto se centra en abordar en profundidad esta última, la cual goza de gran popularidad dentro del sector.



Descripción del problema

Grafana

Grafana es uno de los múltiples software de código abierto que permite la visualización de información almacenada en múltiples bases de datos. Trabaja especialmente bien con BBDD de series de tiempo, tales como Elasticsearch, Graphite, InfluxDB...

Descripción del problema

Grafana



Figure: Ejemplo de dashboard usando Grafana

Descripción del problema

¿Cómo conseguimos los datos?

Es aquí donde entra en juego otra de las herramientas más relevantes de este proyecto, **GrimoireLab**. Gracias a ella, podemos recopilar todos los datos necesarios para después plasmarlos en forma de dashboards.



Descripción del problema

GrimoireLab

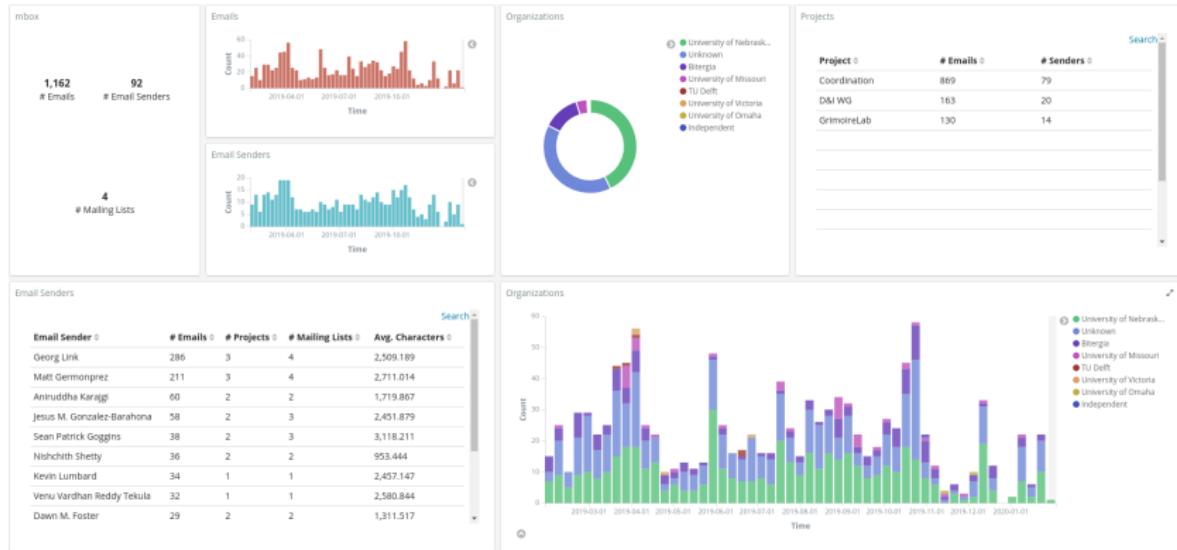


Figure: Dashboard en Kibana, usando los datos de GrimoireLab

Descripción del problema

Integración

Al usuario final le gusta la unificación. Aplicaciones que lo tengan todo: instalación sencilla, fácil de usar, que se integre con otras herramientas, etc. Esto se puede conseguir haciendo uso de **dockers** o contenedores de imágenes, creados gracias a la herramienta Docker Hub.

Descripción del problema

Integración (2)

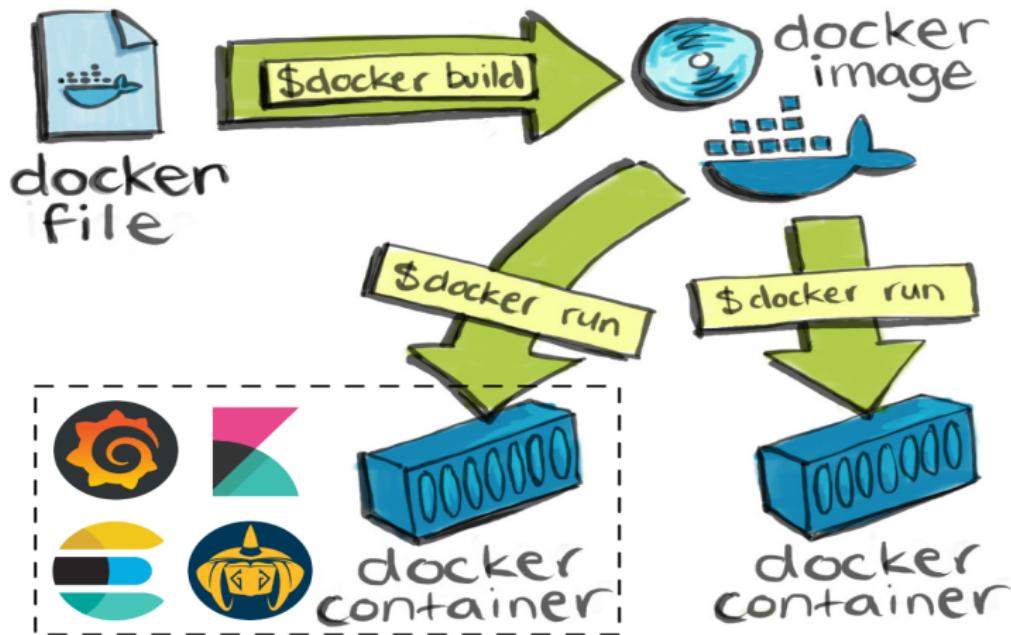


Figure: Unificación mediante dockers

Objetivos principales

- **Representación** de datos con Grafana gracias al uso de GrimoireLab.
- **Unificación** de ambas herramientas en un mismo *docker* gracias a Docker Hub.
- Uso de **multicontenedores**: despliegue de varios dockers de distintos proyectos, ahorrando tiempo en la extracción, enriquecimiento y visualización de datos.
- **Comparación** de las dos herramientas de representación utilizadas: Grafana y Kibana. Para ello se exploran los límites de cada una de ellas (como es el apartado de **plugins** o extensiones de Grafana), señalando además sus semejanzas y diferencias.

Tecnologías utilizadas

① Bases de datos

- Elasticsearch, MariaDB, InfluxDB.

② Sistemas de visualización

- Grafana
- Kibana
- Cauldron

③ Herramientas adicionales

- Git / GitHub
- GrimoireLab
- Docker Hub
- Overleaf

Contenido

① Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

② Desarrollo e implementación

- Metodología Scrum
- Iteraciones o sprints

③ Diseño y resultados

- Arquitectura final
- 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

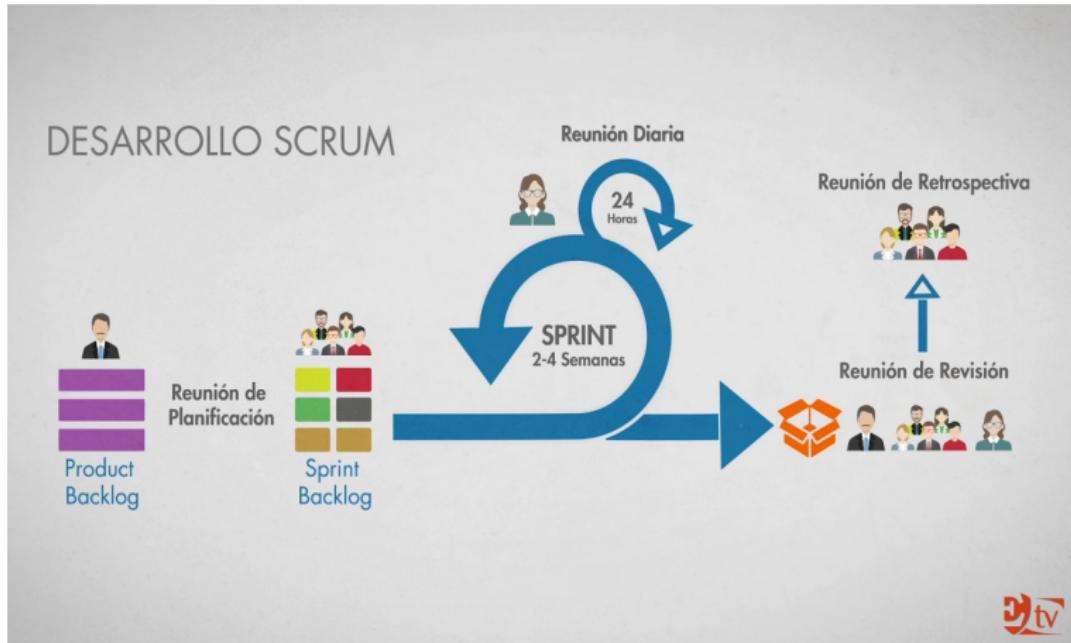


Figure: Esquema del entorno de trabajo Scrum

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

Contenido

① Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

② Desarrollo e implementación

- Metodología Scrum
- Iteraciones o sprints

③ Diseño y resultados

- Arquitectura final
- Evaluación y demostración de la herramienta
- Extensiones o plugins
- Comparativa de herramientas

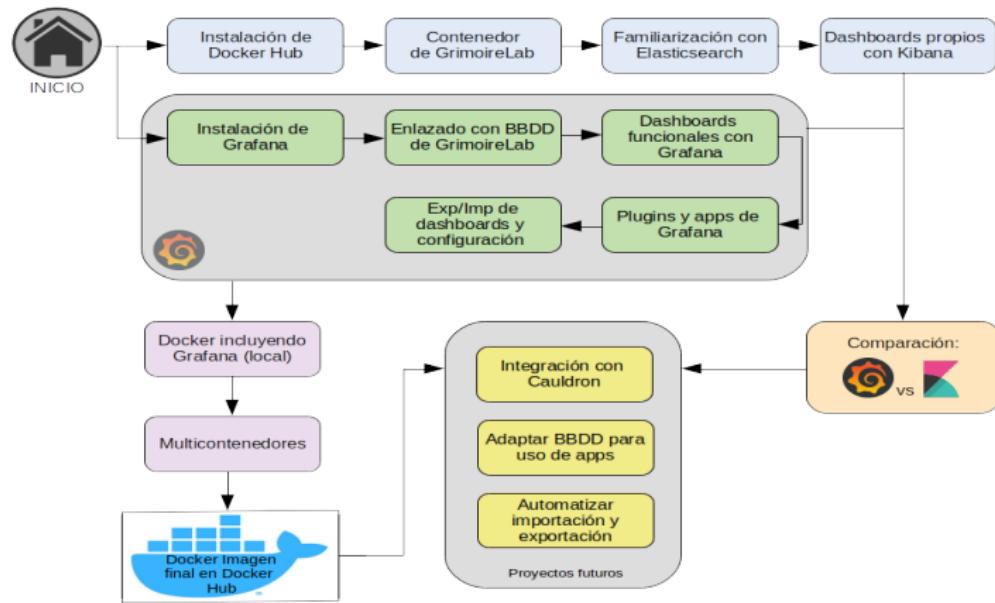
④ Conclusiones

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

⑤ Referencias

Diseño y resultados

Arquitectura final



Diseño y resultados

Demostración de la herramienta

● Git, Mailing Lists, Plugins, Pruebas.

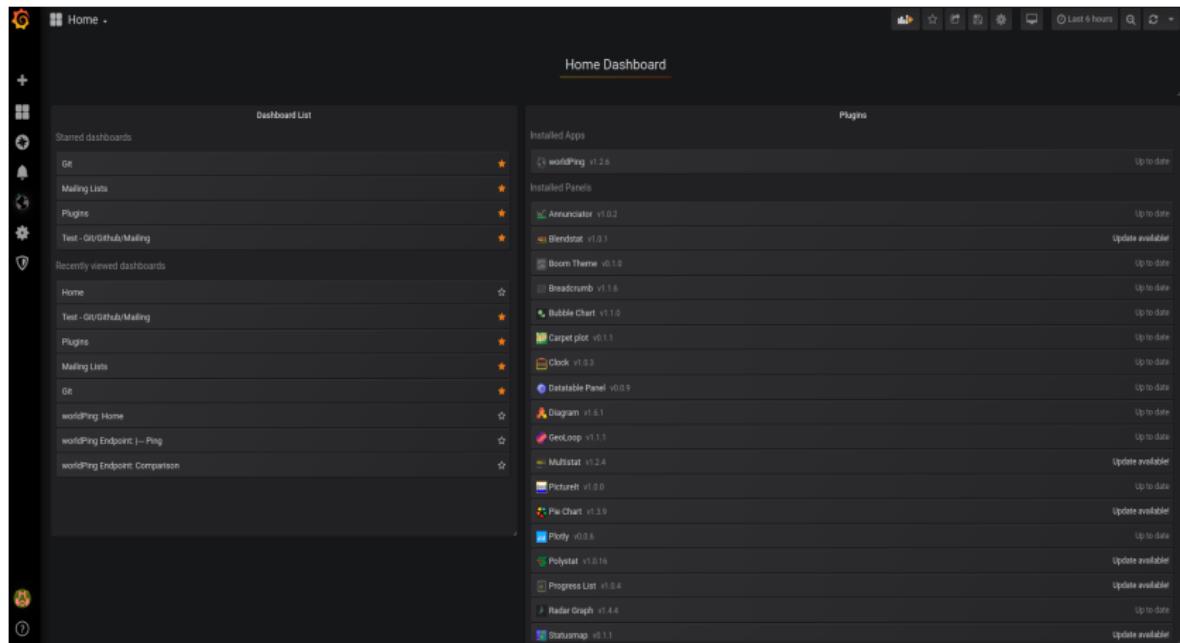


Figure: Dashboard *Home* de Grafana

Diseño y resultados

Demostración de la herramienta (2)

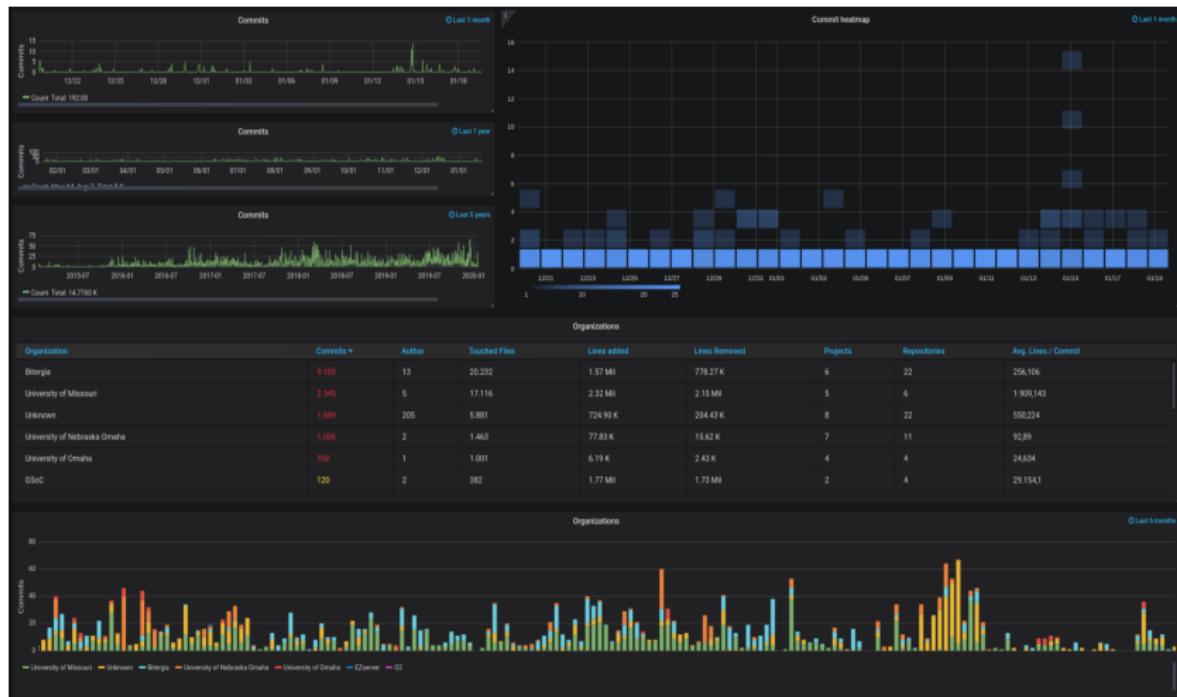


Figure: Extracto del dashboard *Git* realizado con *Grafana*

Diseño y resultados

Extensiones o plugins

Un **plugin**, en este caso, es una simple 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)

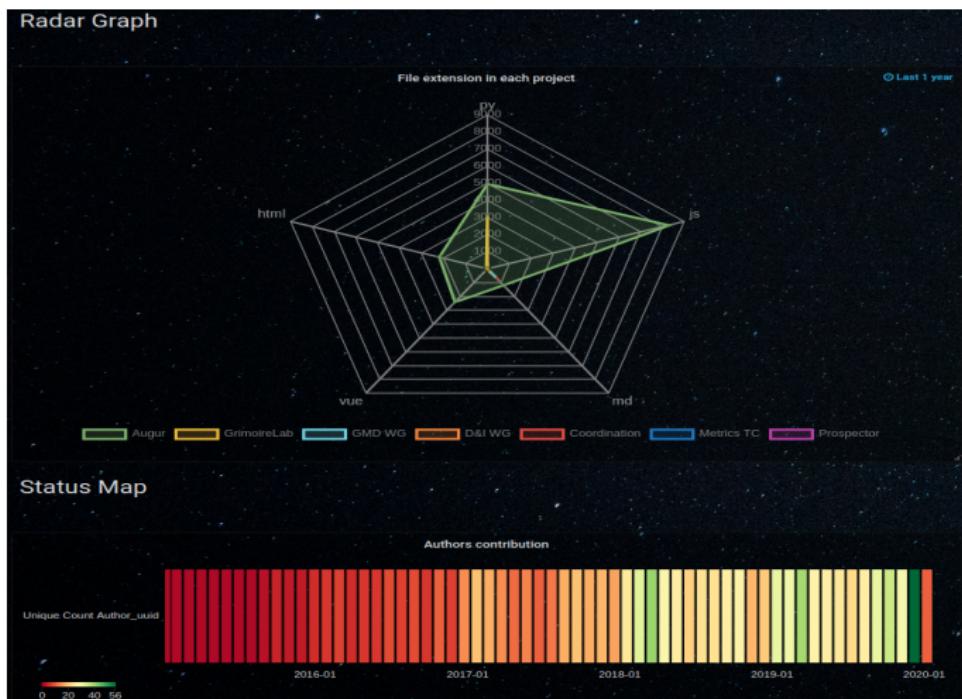


Figure: Ejemplo de plugins (2)

Diseño y resultados

Extensiones o plugins (3)



Figure: Ejemplo de plugins (3)

Diseño y resultados

Comparativa de herramientas

- 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.

Contenido

① Introducción

- Descripción del problema
- Objetivos principales
- Tecnologías utilizadas

② Desarrollo e implementación

- Metodología Scrum
- Iteraciones o sprints

③ Diseño y resultados

- Arquitectura final
- 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

Conclusiones

Consecución de objetivos

- **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. ✓
- **Unificación** de las herramientas en un solo docker¹ o contenedor. ✓
- Creación de contenedores adicionales para el despliegue de **múltiples proyectos**. ✓
- **Comparación**, mencionando numerosas ventajas/desventajas y similitudes/diferencias entre las dos herramientas (Grafana vs Kibana). ✓

¹<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.

Conclusiones

Trabajos futuros

- Desplegar esta herramienta en la nube, en algún servidor de internet. Posibilidad de **integrar** esta aplicación en un proyecto de mayor envergadura (**Cauldron**²).
- 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.

²<https://gitlab.com/cauldronio/cauldron-deployment>

Referencias y bibliografía

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 y docker image

- Web del proyecto.

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

- Docker image del proyecto.

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