Despliegue automatizado y monitoreo de Ghost CMS mediante contenedores Docker

Integrante:

Jonathan Franco Santana

Tutor:

Julio César Gómez

Corporación Colsubsidio Educación Tecnológica (CET)

en alianza con la Universidad de los Andes

Curso en Administración de Servidores y Virtualización

13 de diciembre de 2025

Contenido

Resumen

El presente proyecto integrador tiene como objetivo aplicar de manera práctica los conocimientos adquiridos en la gestión de servidores, virtualización, automatización y monitoreo de servicios. Para ello, se seleccionó Ghost CMS, una plataforma de gestión de contenidos orientada a la creación de blogs y boletines digitales, la cual fue desplegada en un entorno virtualizado mediante contenedores Docker.

El despliegue del servicio fue automatizado mediante scripts en PowerShell, permitiendo la inicialización de Docker Desktop, el levantamiento de los servicios y su correcta operación de forma controlada. Adicionalmente, se integró un sistema de monitoreo basado en Prometheus y Grafana, el cual permite observar en tiempo real el estado de la infraestructura y el consumo de recursos de los contenedores.

Los resultados obtenidos evidencian una correcta implementación técnica del servicio, un alto nivel de automatización y un monitoreo funcional, cumpliendo así con los objetivos planteados en el curso.

Introducción

La administración moderna de servidores requiere el uso de herramientas que permitan desplegar servicios de forma eficiente, reproducible y escalable. La virtualización mediante contenedores ha ganado gran relevancia debido a su ligereza y facilidad de automatización, permitiendo simular entornos reales de producción.

En este contexto, el presente proyecto se desarrolla en el marco del Curso en Administración de Servidores y Virtualización, impartido por la Corporación Colsubsidio Educación Tecnológica (CET) en alianza con la Universidad de los Andes, con el fin de fortalecer competencias prácticas en el despliegue, automatización y monitoreo de servicios.

Objetivo general

Implementar y automatizar el despliegue de un servicio web funcional utilizando contenedores Docker, integrando un sistema de monitoreo que permita supervisar el estado de la infraestructura y del servicio en ejecución.

Objetivos específicos

- Seleccionar un servicio en línea funcional y justificar su elección.
- Diseñar una arquitectura basada en contenedores.
- Automatizar el despliegue de la infraestructura y del servicio.
- Implementar un sistema de monitoreo utilizando Prometheus y Grafana.
- Documentar el proceso y validar el correcto funcionamiento del sistema.

Selección del servicio

Servicio seleccionado: Ghost CMS

Ghost CMS es una plataforma de gestión de contenidos enfocada en la publicación de blogs y boletines informativos. Se caracteriza por su arquitectura moderna, su facilidad de despliegue y su compatibilidad con contenedores Docker y bases de datos MySQL.

Justificación de la elección

La selección de Ghost CMS se realizó teniendo en cuenta los siguientes criterios:

- Es un servicio real y ampliamente utilizado en entornos productivos.
- Cuenta con documentación oficial completa.
- Permite una integración sencilla con bases de datos.
- Es ideal para demostrar procesos de automatización y monitoreo.

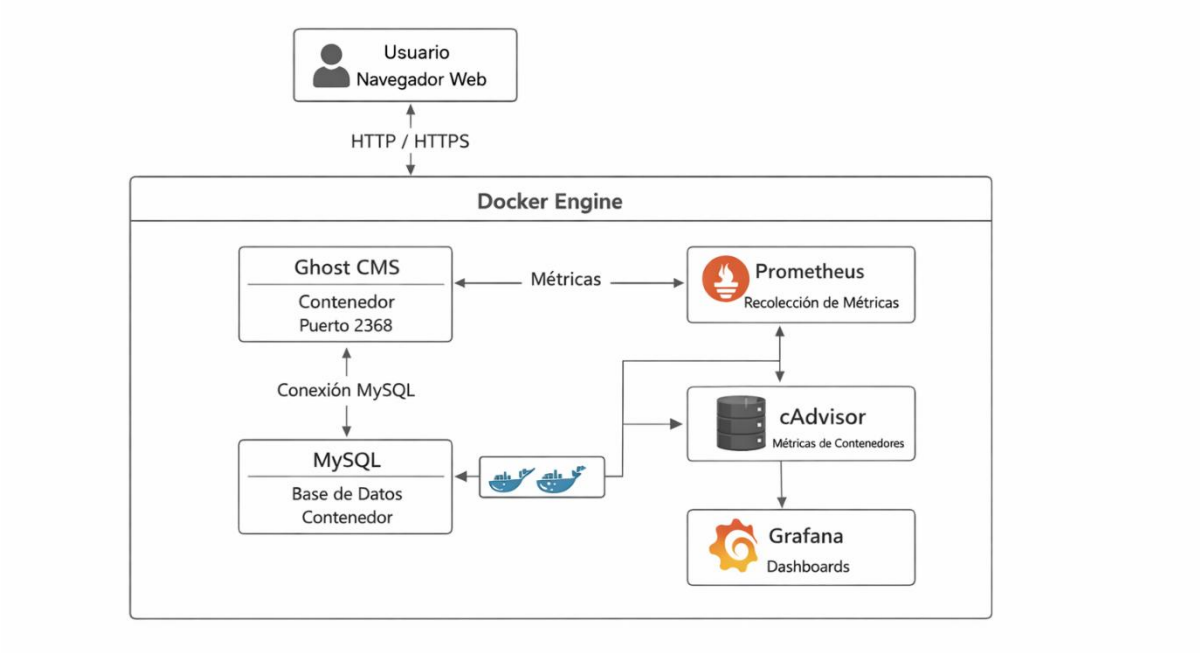
Diseño de la arquitectura

La arquitectura del proyecto se basa en contenedores Docker orquestados mediante Docker Compose. Los componentes principales son:

- Contenedor de Ghost CMS.
- Contenedor de base de datos MySQL.
- Contenedor de phpMyAdmin para administración de la base de datos.

- Contenedor de Prometheus para recolección de métricas.
- Contenedor de Grafana para visualización de métricas.
- Contenedor de cAdvisor para monitoreo de contenedores.

Figura 1. Diagrama de arquitectura del sistema



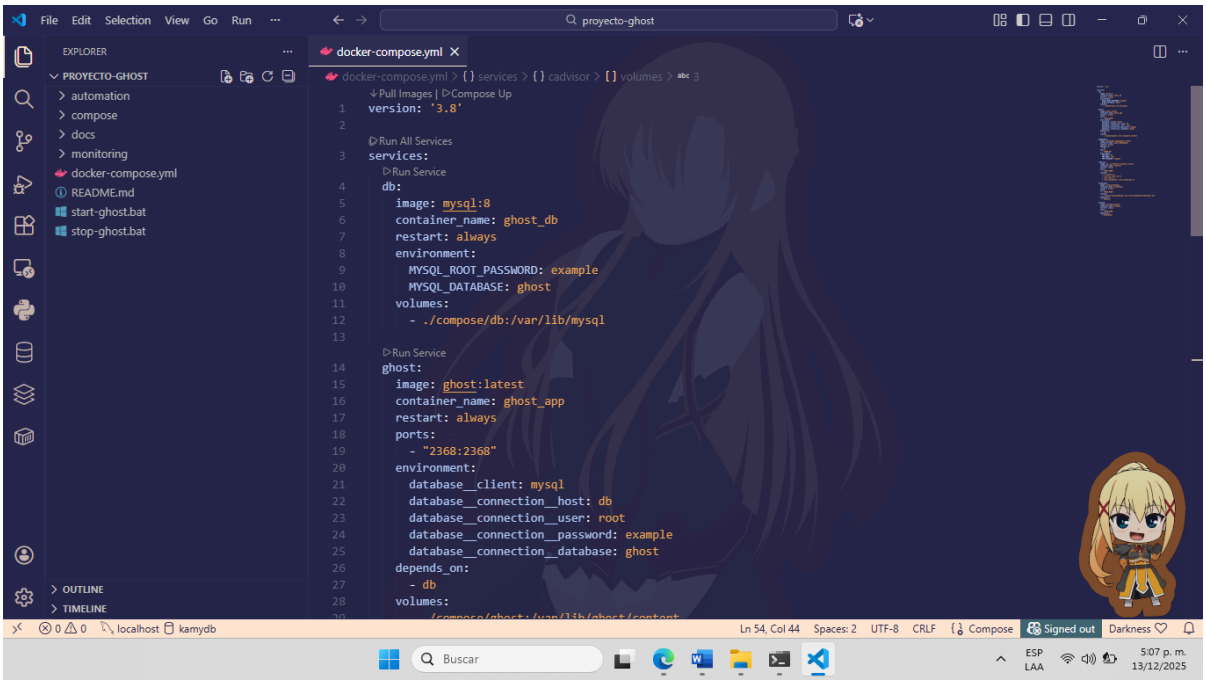
Nota. *Elaboración propia (2025). Disponible en Google Drive*

Contenerización y despliegue

El despliegue del servicio se realizó utilizando Docker y Docker Compose, definiendo todos los servicios en un archivo docker-compose.yml. Esta configuración permite levantar toda la infraestructura con un solo comando.

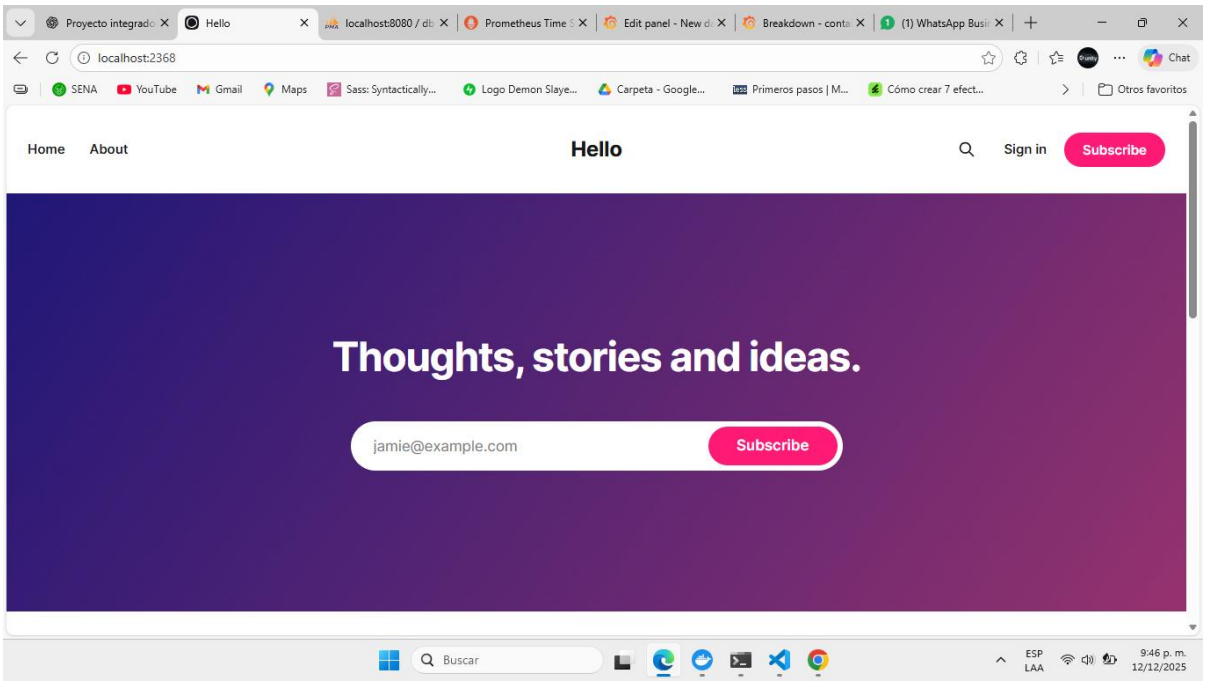
El servicio Ghost quedó accesible de manera local, permitiendo la validación de su correcto funcionamiento tanto en el frontend como en el panel administrativo.

Figura 2. Archivo docker-compose.yml



Nota. *Elaboración propia (2025). Disponible en Google Drive*

Figura 3. Ghost CMS en funcionamiento



Nota. *Elaboración propia (2025). Disponible en Google Drive*

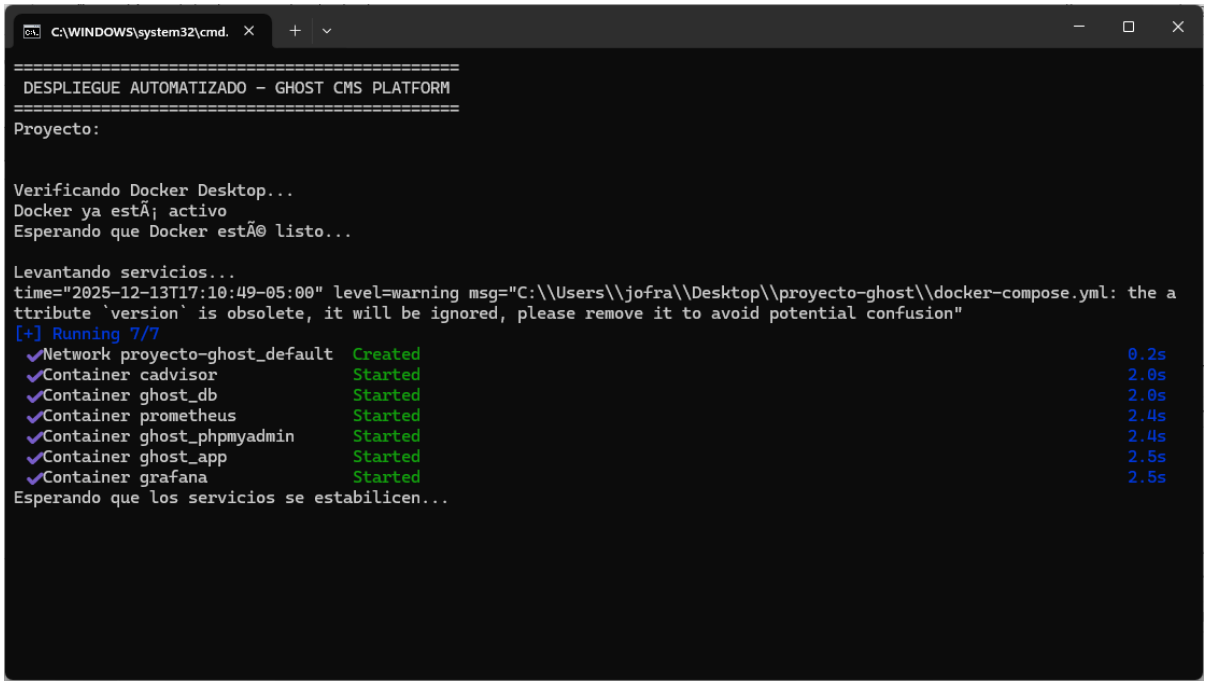
Automatización del despliegue

La automatización se implementó mediante scripts en PowerShell, los cuales permiten:

- Verificar el estado de Docker Desktop.
- Iniciar Docker en caso de que no esté activo.
- Levantar los servicios definidos en Docker Compose.
- Validar que los contenedores se encuentren en ejecución.

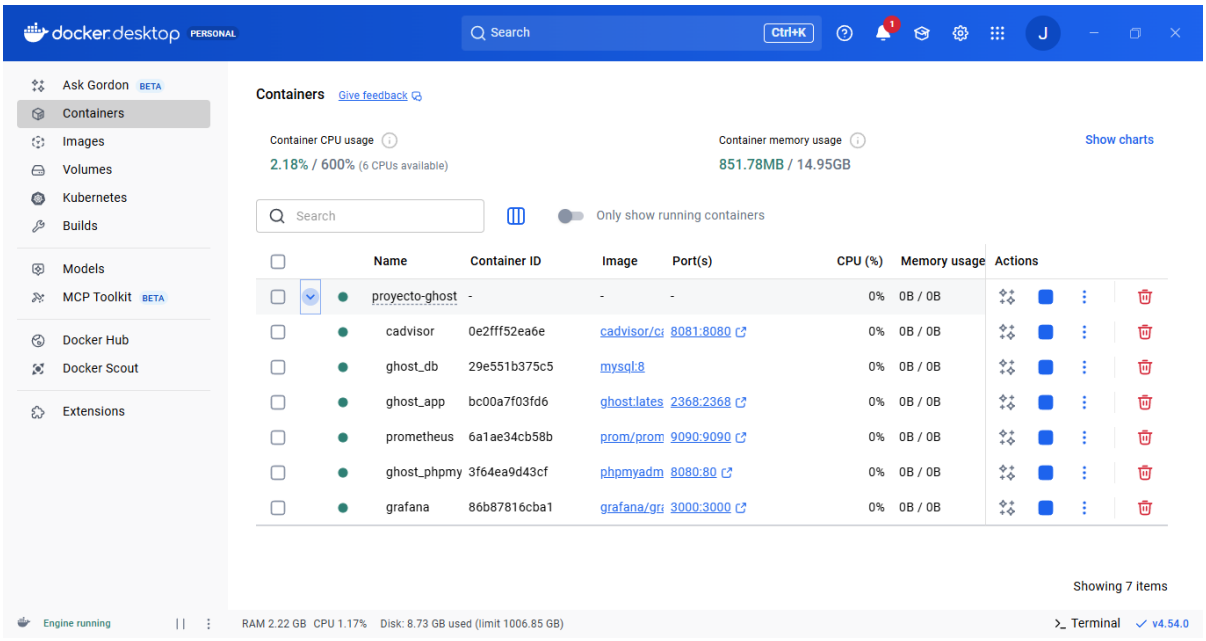
Este proceso reduce la intervención manual y simula un entorno automatizado similar al de producción.

Figura 4. Ejecución del script de automatización



Nota. *Elaboración propia (2025). Disponible en Google Drive*

Figura 5. Contenedores activos



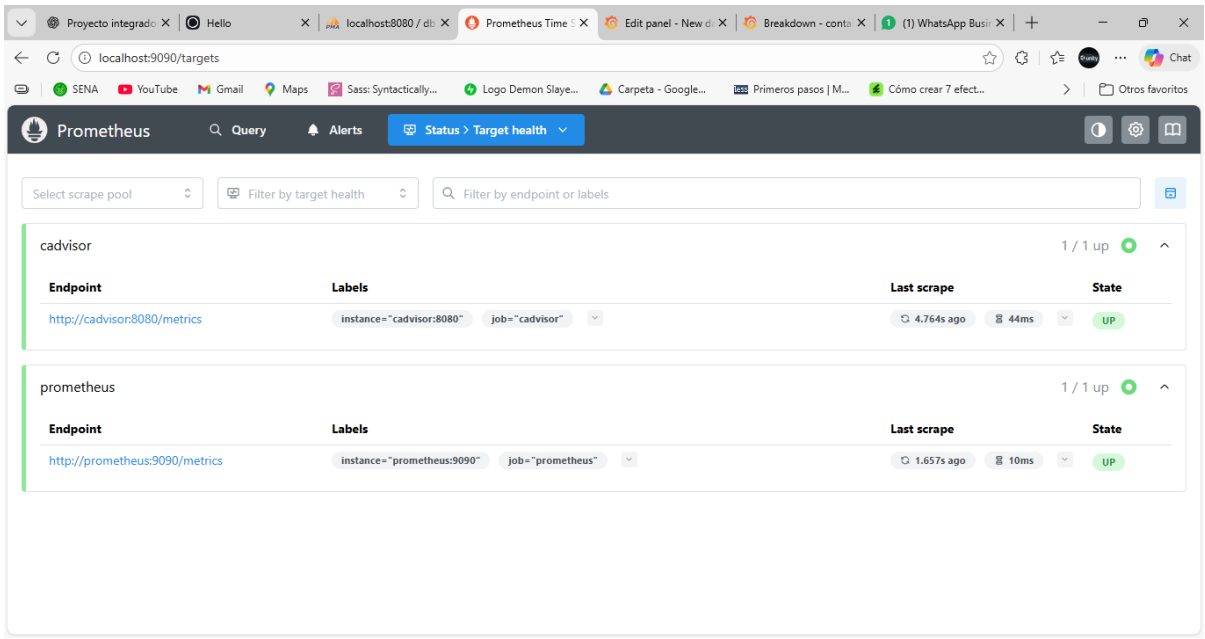
Nota. *Elaboración propia (2025). Disponible en Google Drive*

Monitoreo del sistema

Prometheus

Prometheus se utilizó para la recolección de métricas relacionadas con el consumo de CPU, memoria y estado de los contenedores. Este sistema permite supervisar el rendimiento general de la infraestructura.

Figura 6. Prometheus en ejecución

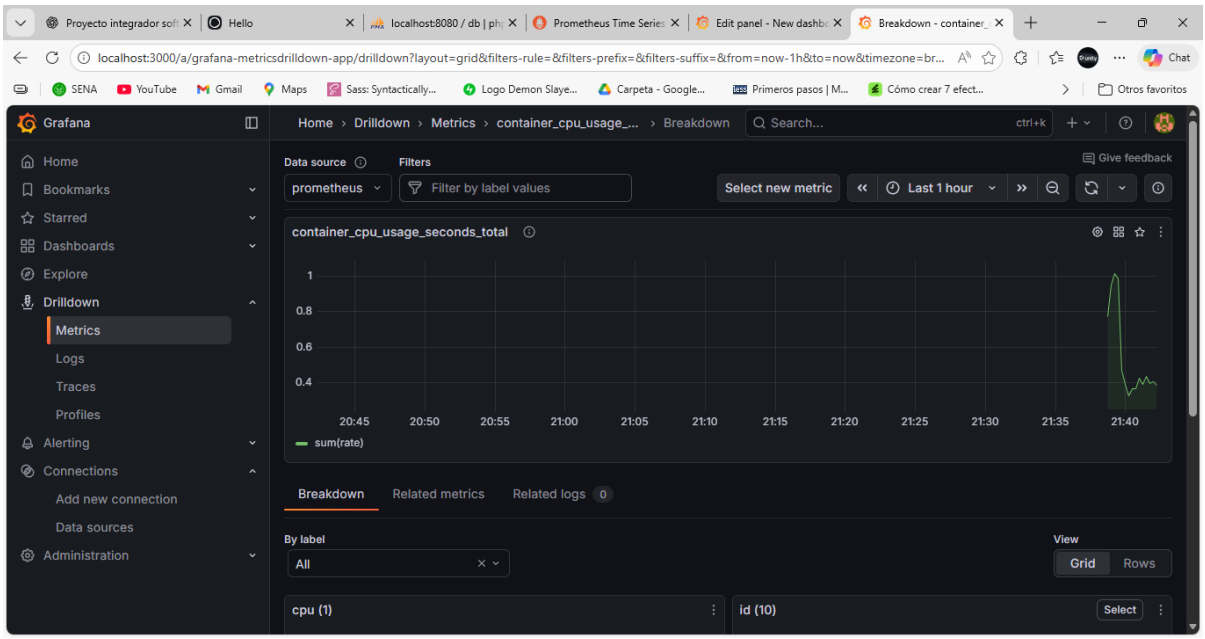


Nota. *Elaboración propia (2025). Disponible en Google Drive*

Grafana

Grafana se integró para la visualización de métricas a través de dashboards, facilitando la interpretación de la información recolectada por Prometheus.

Figura 7. Dashboard de Grafana



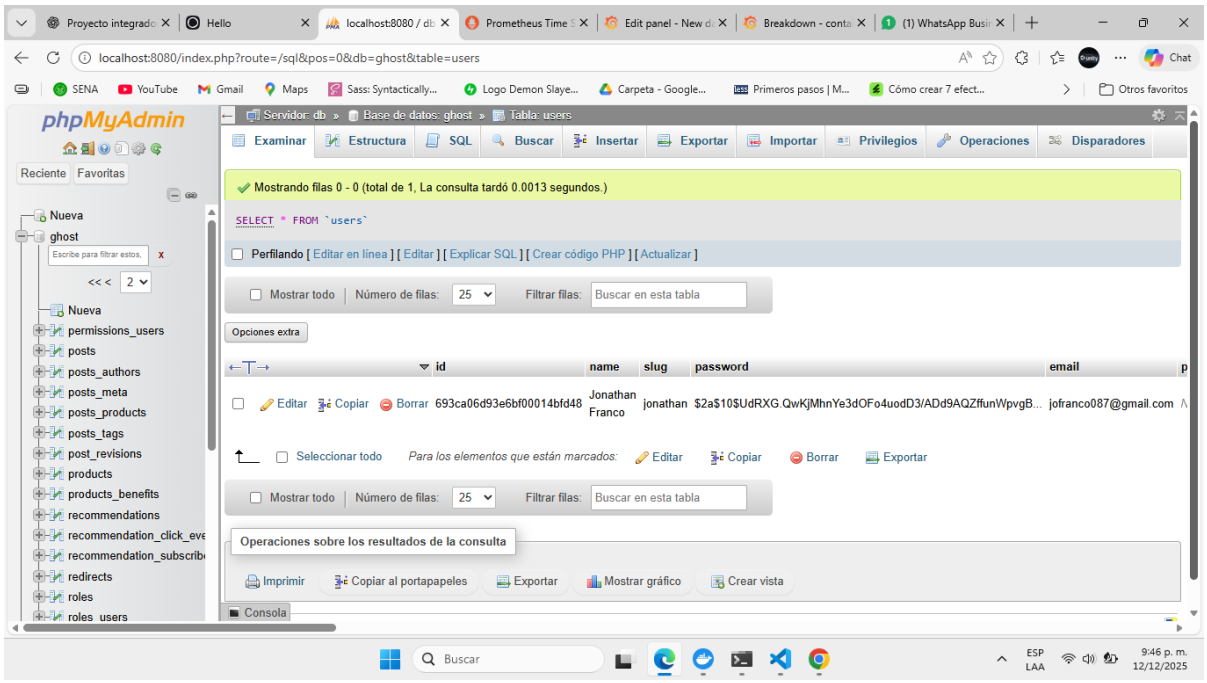
Nota. *Elaboración propia (2025). Disponible en Google Drive*

Validación del sistema

Se realizaron pruebas de acceso a todos los servicios desplegados, confirmando su correcto funcionamiento:

- Acceso a Ghost CMS.
- Acceso al panel administrativo.
- Conexión a la base de datos mediante phpMyAdmin.
- Visualización de métricas en Grafana.

Figura 8. phpMyAdmin conectado a la base de datos



Nota. *Elaboración propia (2025). Disponible en Google Drive*

Retos y soluciones

Tabla. 1

Retos identificados y soluciones implementadas en el despliegue del sistema

Reto identificado	Solución implementada
Inicio manual de Docker	Automatización con PowerShell
Monitoreo de contenedores	Integración de cAdvisor
Gestión de múltiples servicios	Orquestación con Docker Compose

Nota. *Elaboración propia (2025).*

Conclusiones

El desarrollo de este proyecto permitió aplicar de manera práctica los conceptos de virtualización, automatización y monitoreo de servicios. La implementación de Ghost CMS mediante contenedores Docker demostró ser una solución eficiente y flexible.

La automatización del despliegue redujo significativamente la complejidad operativa, mientras que el monitoreo con Prometheus y Grafana proporcionó una visión clara del estado del sistema. En conclusión, el proyecto cumple satisfactoriamente con los objetivos planteados y con los criterios de evaluación establecidos.

Referencias

Docker Inc. (2024). *Docker documentation*. https://docs.docker.com

Ghost Foundation. (2024). *Ghost documentation*. https://ghost.org/docs

Prometheus Authors. (2024). *Prometheus documentation*. https://prometheus.io/docs

Grafana Labs. (2024). *Grafana documentation*. https://grafana.com/docs