

MSIN - 4215: Seguridad en Cloud
Proyecto I: Despliegue de aplicaciones en Cloud

1 Introducción

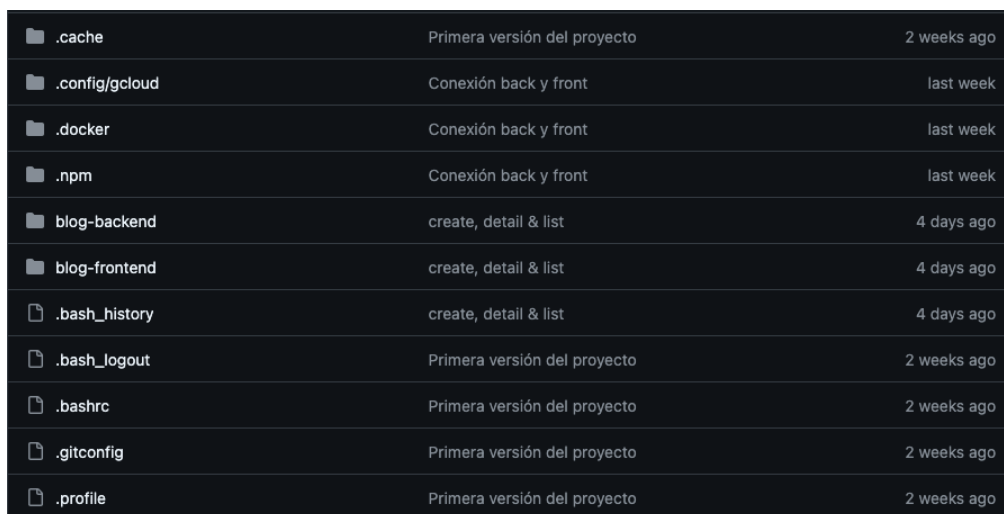
Este proyecto consiste en el diseño, desarrollo y despliegue de una plataforma simplificada de blogging. Se utilizará FastAPI para el backend, una tecnología de frontend de libre elección por parte del estudiante, y la infraestructura de Google Cloud Platform (GCP).

La plataforma permitirá a los usuarios crear, editar, publicar, eliminar y categorizar entradas de blog mediante etiquetas (tags), además de gestionar la calificación de las publicaciones por parte de los lectores. El enfoque principal estará en la construcción de una API RESTful robusta y segura, la interacción eficiente entre el frontend y el backend, y el despliegue en un entorno real de nube.

La aplicación deberá estar desplegada en la nube de GCP y en ejecución sobre contenedores Docker basados en GNU/Linux. Debe incluir Dockerfiles para la generación de las imágenes y Docker Compose para el despliegue y ejecución la aplicación en su totalidad.

2 Estructura del proyecto

En el caso de este proyecto, estamos trabajando en un solo repositorio en GitHub. Esto no necesariamente pasaría en un proyecto de desarrollo hecho para fines empresariales, sin embargo, dado que en este caso nuestras intenciones son aquellas de mantener una aplicación de carácter mínimo, dicha estructura nos favorece, por lo menos por ahora.



📁 .cache	Primera versión del proyecto	2 weeks ago
📁 .config/gcloud	Conexión back y front	last week
📁 .docker	Conexión back y front	last week
📁 .npm	Conexión back y front	last week
📁 blog-backend	create, detail & list	4 days ago
📁 blog-frontend	create, detail & list	4 days ago
📄 .bash_history	create, detail & list	4 days ago
📄 .bash_logout	Primera versión del proyecto	2 weeks ago
📄 .bashrc	Primera versión del proyecto	2 weeks ago
📄 .gitconfig	Primera versión del proyecto	2 weeks ago
📄 .profile	Primera versión del proyecto	2 weeks ago

Figure 1: Estructura de archivos del proyecto

En la carpeta 'blog-backend' se encuentra una versión del backend implementado a partir del framework de FastAPI. En la carpeta 'blog-frontend' tenemos una aplicación web, montada en el framework 'React'. Ambas carpetas tienen imágenes Docker asociadas, las cuáles se encargarán de las dependencias del lenguaje de programación del proyecto. A la hora de desplegarlo a nube, usaremos máquinas virtuales basadas en Ubuntu, que se encargarán del resto del sistema bajo el cuál se ejecuta. Esto nos permitió a la hora de trabajar con el proyecto mantener una cantidad mínima de dependencias necesarias para su ejecución. No solo esto no nos ata a más tecnologías de las necesarias, sino que hace el desarrollo de la aplicación más flexible, dado que tiene menos implicaciones en memoria correr estas imágenes sobre aquellas que incluyen sistemas operativos. Dado esto, también se podría correr el sistema en máquinas virtuales más baratas.

2.1 Tecnologías utilizadas

- Python: Para la infraestructura del backend.
- TypeScript/React: para el código del frontend.
- Docker: para poder containerizar la aplicación.
- Google Cloud Platform (GCP): para desplegar en nube la solución

2.2 Instrucciones de Ejecución

Para ejecutar la aplicación, sea on-premise o en una máquina virtual, vamos a ejecutar la imagen en docker, y vamos a utilizar una plantilla de un entorno virtual de Python incluida dentro del entorno de backend. Esto se ofrece principalmente porque en un entorno on-premise va a separar los datos de librerías que la aplicación contiene de aquellos que son propios de la máquina.

Dicho esto, el primer paso es clonar el proyecto desde github, de la siguiente manera:

```
git clone https://github.com/MariaLuisaR/BlogCloud.git
```

una vez hecho esto, activaremos el 'venv' y la imagen docker.

```
cd blog-backend
source venv/bin/activate
docker compose down --volumes --remove-orphans
docker compose up -d --build
```

Esto va a abrir un puerto y una dirección en la VM o el computador del que se ejecuta este sistema, entonces, ahora para acceder al sitio, vamos ir a dicha dirección. La aplicación corre actualmente en la ip publica de la máquina virtual 34.28.37.200. El puerto del front es 3000, por lo cual se puede acceder a el en la siguiente dirección: <http://ip-publica:3000/> (<http://34.28.37.200:3000/> en la máquina virtual propia)

2.3 Escalabilidad

Docker esta pensado con la idea de generar entornos replicables, y de esta misma forma, el docker proporcionado esta pensado para que se pueda establecer el mismo entorno en varias máquinas. Junto con este, nuestra configuración de GCP tiene una maquina base, la cuál nos permite mantener las variables concernientes al sistema operativo en mente. Por esto mismo, podemos establecer un esquema de escalabilidad horizontal, en el cuál podríamos rápidamente montar maquinas virtuales que repliquen este mismo entorno. Mientras que todas esten conectadas a una misma base de datos (por ejemplo, Cloud SQL) deberíamos poder asumir que dada la simplicidad de esta solución podría acceder a un mismo estado sin muchos problemas.

2.4 Seguridad

2.4.1 Manejo de contraseñas

A la hora de obtener la contraseña de un usuario en el backend, el sistema no guarda las contraseñas en texto plano, sino que utiliza un hash para guardar la contraseña. Esto permite evitar que, en caso de un filtrado de información en nuestra aplicación y dadas contraseñas lo suficientemente robustas de parte de nuestros usuarios, un filtrado de sus datos resulte menos grave de lo que podría resultar de lo contrario.

2.4.2 Depuración para SQL

Parte de la razón por la que se eligieron las tecnologías que se eligieron es por la implementación de infraestructura previa en todas estas contra la ejecución de ataques SQL. El sistema de envío de variables de JSX, el cuál incluye un sistema de depuración de inputs, que básicamente convierte en cadenas de caracteres el input del usuario en tiempo de ejecución. Esto básicamente es una medida inicial contra la ejecución de código y el XSS. Sin embargo, es importante recalcar que esto también tiene la implicación de que cualquier vulnerabilidad encontrada en este sistema externo nos afectará a nosotros también. Dado esto, en un entorno de ejecución real, sería imperativo para el mantenimiento de la aplicación mantener su infraestructura debidamente actualizada.

2.4.3 Entorno IAM

en el entorno de despliegue GCP, existe un sistema de verificación IAM para poder entrar a los datos propuestos dentro de Cloud SQL. Esto nos permite evitar que accesos indeseados a la base de datos ocurran sin comprometer una cuenta de administración (que para efecto de este laboratorio, seria aquella de los 2 desarrolladores)