



ESCUELA DE  
INGENIERÍA EN CIENCIAS Y SISTEMAS  
FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



**Día, Fecha:**

Miércoles, 05/10/2022

**Hora de inicio:**

15:40

# Sistemas Operativos 1 [Sección N]

German José Paz Córdón

# Clase 10



# Información



- Tutor : German José Paz Cordon
- Correo : [2776871601601@ingenieria.usac.edu.gt](mailto:2776871601601@ingenieria.usac.edu.gt)
- Asunto : [SO1]<<Duda | Consulta | etc>>
- [https://github.com/GermanJosePazCordon/Laboratorio\\_SO1\\_2S22](https://github.com/GermanJosePazCordon/Laboratorio_SO1_2S22)

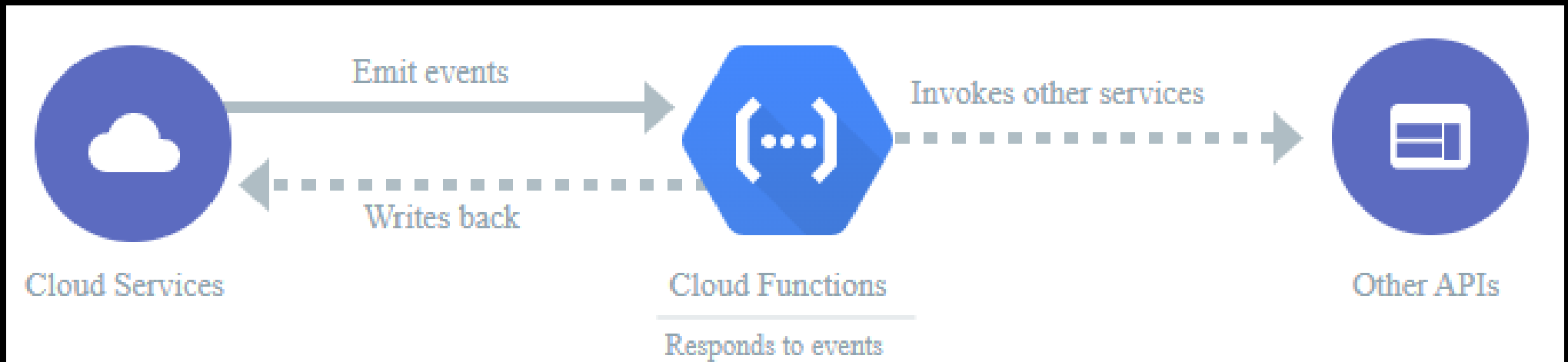
# Agenda

- Foro 10
- Notas, Grupos
- Proyecto
- Ejemplo Práctico





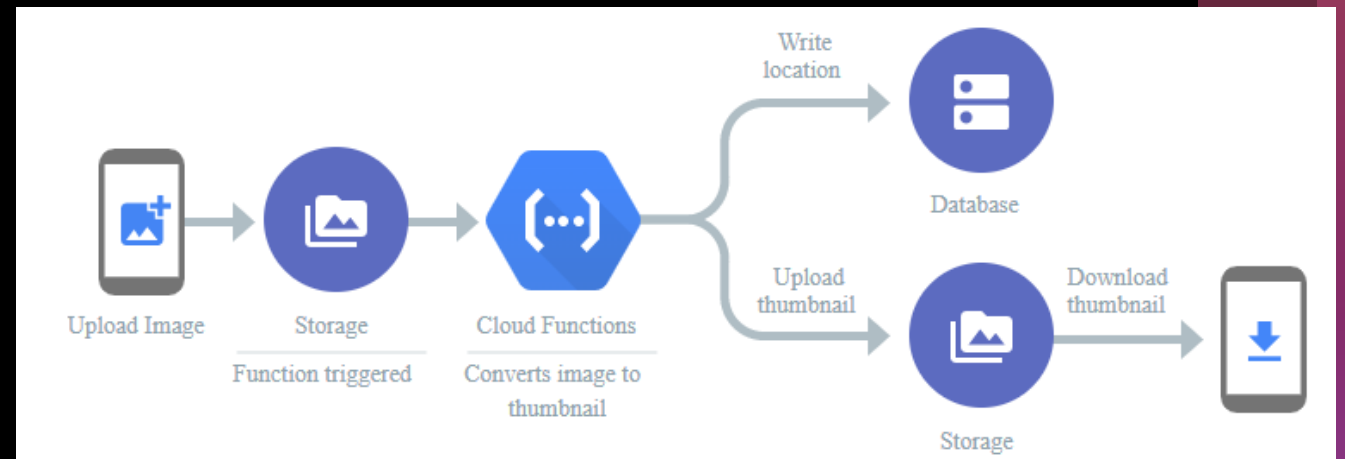
# Cloud Functions



- Sirve para crear aplicaciones sin servidores dentro de la infraestructura de Google que den respuesta a la demanda de eventos que puedan ocurrir en cualquier lugar.
- Lo bueno de este servicios es que se paga únicamente lo que se usa, es decir el tiempo que el código se esté ejecutando, de manera que es ideal para soluciones pequeñas.

# Ejemplos de desarrollos

- Backends de aplicaciones sin servidor.
- Procesamiento de datos en tiempo real.
- Aplicaciones inteligentes.





**Locust**



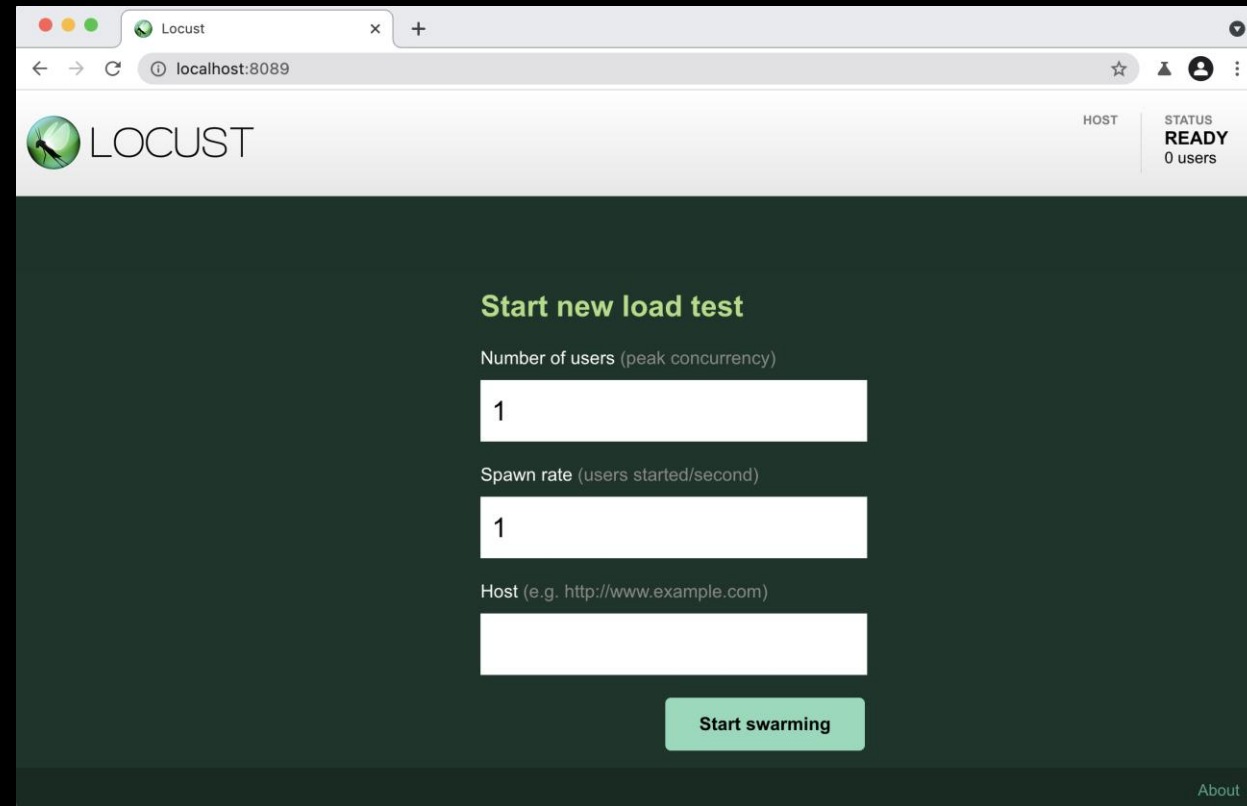


# LOCUST

- Locust es una herramienta de prueba de carga de código abierto. La prueba de carga es un tipo de prueba de software que se realiza para verificar la tolerancia o el comportamiento del sistema bajo una carga esperada específica.

# Características

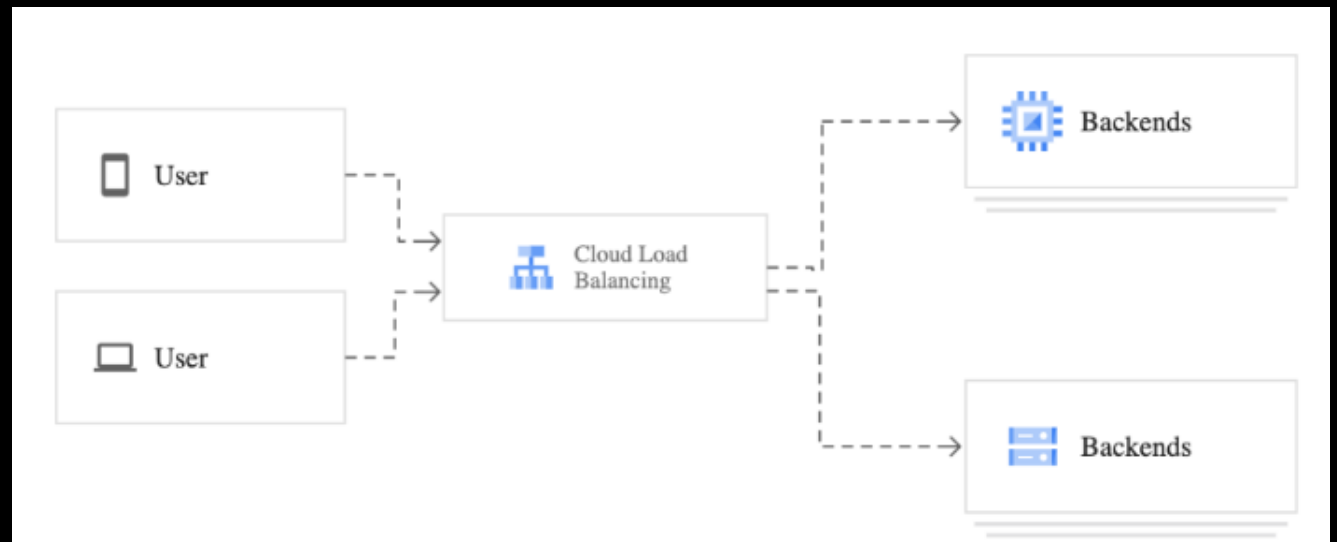
- Los escenarios de prueba se pueden escribir en Python.
- Distribuido y escalable.
- UI basada en web.
- Cualquier sistema se puede probar con esta herramienta.





# Load Balancer

- Load Balancer o Balanceador de Carga, se refiere a la distribución del tráfico de red entrante a través de un grupo de servidores backend.
- El balance de carga enruta las solicitudes de los clientes en todos los servidores para satisfacer esas solicitudes de manera que maximice la velocidad y la capacidad para poder garantizar que ningún servidor esté sobrecargado, ya que la saturación podría afectar el rendimiento .



# Fuciones



- Distribuye las solicitudes de los clientes o la carga de la red de manera eficiente en varios servidores.
- Asegura alta disponibilidad y confiabilidad al enviar solicitudes solo a servidores que están en línea.
- Proporciona la flexibilidad de agregar o restar servidores según la demanda.



**Gracias por su atención**