

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
SISTEMAS OPERATIVOS 1 SECCIÓN N  
ING. JESUS GUZMAN POLANCO  
AUX. JOSÉ DANIEL VELÁSQUEZ OROZCO  
AUX. JHONATHAN DANIEL TOCAY  
SEGUNDO SEMESTRE 2023



## PROYECTO 2

# Sistema de Registro de Notas

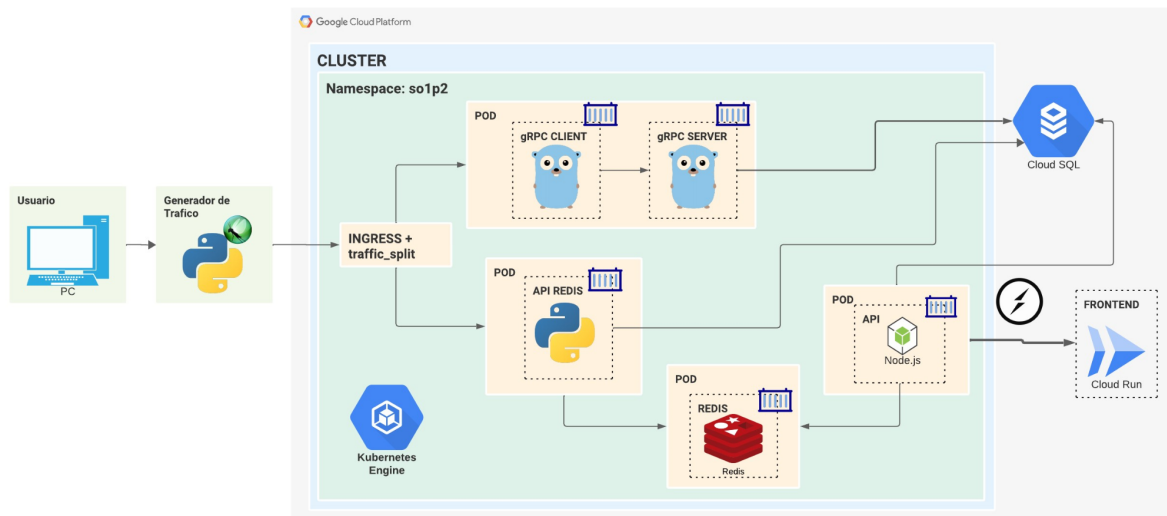
### Objetivos

- Desplegar un clúster de Kubernetes y sus servicios.
- Conocer ServerLess por medio de Cloud Run.
- Utilizar bases de datos en la nube con Cloud SQL y Memory Store.
- Controlar el tráfico de datos por medio de balanceadores de carga.

### Introducción

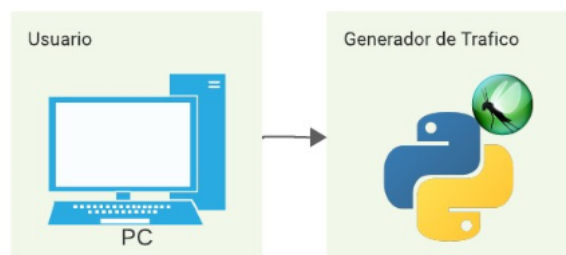
El objetivo de este proyecto es evaluar y mejorar el rendimiento de un sistema de ingreso de notas en línea mediante pruebas de carga y rendimiento utilizando Locust, Kubernetes y servicios en la nube como Cloud SQL y Cloud run. El sistema permite a los profesores o administradores académicos ingresar y gestionar las calificaciones de los estudiantes de manera eficiente. Además se debe de implementar una Aplicación web donde se muestre las estadísticas y reportes de las notas de los estudiantes.

# Arquitectura



## Generador de Trafico

Esta parte consiste en la creación de un generador de tráfico de entrada utilizando Locust y Python como lenguaje de programación. El tráfico será recibido en un Ingress y dividido con un traffic split. El contenido deberán ser los parámetros para enviar aleatoriamente en cada solicitud, estos datos se obtendrán de manera aleatoria.



## Configuración de Locust

Se debe de generar un scripts de prueba Locust para simular el comportamiento de usuarios ingresando calificaciones. El JSON de Prueba tendrá un formato de entrada similar al siguiente:

```
{
  "carnet": 233233,
  "nombre": "Alumno 1",
  "curso": "SO1",
  "nota": 90,
  "semestre": "2S",
  "year": 2023
}
```

Los Cursos a Monitorear pueden ser:

- **SO1** - Sistemas Operativos 1
- **BD1** - Sistemas de Bases de Datos 1
- **LFP** - Lenguajes Formales y de Programación
- **SA** - Software Avanzado

- **AYD1** - Análisis y Diseño 1

La descripción para semestre puede ser:

- **1S** - Primer Semestre.
- **2S** - Segundo Semestre.

**Año:** 2023

**Nota:** 50, 60, 70, 80, 90, 100.

## Kubernetes

Se encargará de la orquestación de los contenedores de las diferentes partes de la aplicación. Se debe de crear un Namespace llamado: **so1p2**, y trabajar dentro de él para crear los **PODS** y los Servicios.

## Ingress

El Ingress se encargará de gestionar el acceso externo a los servicios dentro del clúster. Actúa como una puerta de entrada que posibilita la exposición de servicios HTTP y HTTPS fuera del clúster. Por otro lado, el Traffic Split se encargará de distribuir de manera equitativa el tráfico de datos. En otras palabras, la configuración del tráfico será del 50% para la ruta 1 y del 50% para la ruta 2.

## Rutas

En esta sección se centra en desarrollar un método altamente eficiente para registrar datos en MySQL a través de comunicación RPC (Remote Procedure Call). El propósito es evaluar y contrastar el rendimiento con respecto a otro enfoque existente. La implementación de la primera estrategia implica recibir los datos de entrada mediante el cliente gRPC, posteriormente transmitir esta información al servidor gRPC y, por último, almacenar los datos en la base de datos de MYSQL que está en el servicio de CLOUD SQL. El Flujo de la segunda ruta se inserta en Memorystore y también hacia CLOUD SQL.

### Ruta 1

La Primera Ruta será desarrollada en GOLANG, estructura:

1. grpc-client
  - a. Recibe la información y la envía al server.
2. grpc-server
  - a. Escribe en la base de datos de MySQL.

### Ruta 2

La Segunda Ruta será desarrollada en Python, estructura:

1. API REDIS:
  - a. Escribir en la base de datos de Redis.
  - b. Escribir en la base de datos de MySQL.

## Autoscaling de k8s (Horizontal Pod Autoscaling)

El HPA ajusta automáticamente la cantidad de PODS en ejecución en un despliegue o conjunto de réplicas según la utilización de la CPU u otras métricas. Para el Proyecto 2, cada Pod debe tener un mínimo de 1 réplica y un máximo de 3 réplicas. Además, se debe configurar que el uso de la CPU no exceda el 50%

## SOCKET IO

Se debe implementar un canal de websockets para conexión en tiempo real entre el frontend y el backend hacia (REDIS), de esta manera el consumo de recursos estará optimizada para gran cantidad de información.

## CLOUD SQL

Es un servicio de base de datos relacional totalmente administrado. Proporciona un almacenamiento eficiente y escalable para los datos del sistema de calificaciones. Para este Proyecto se debe de Crear una Base de Datos Mysql, el modelo de la base de datos queda a discreción del estudiante.

## CLOUD RUN

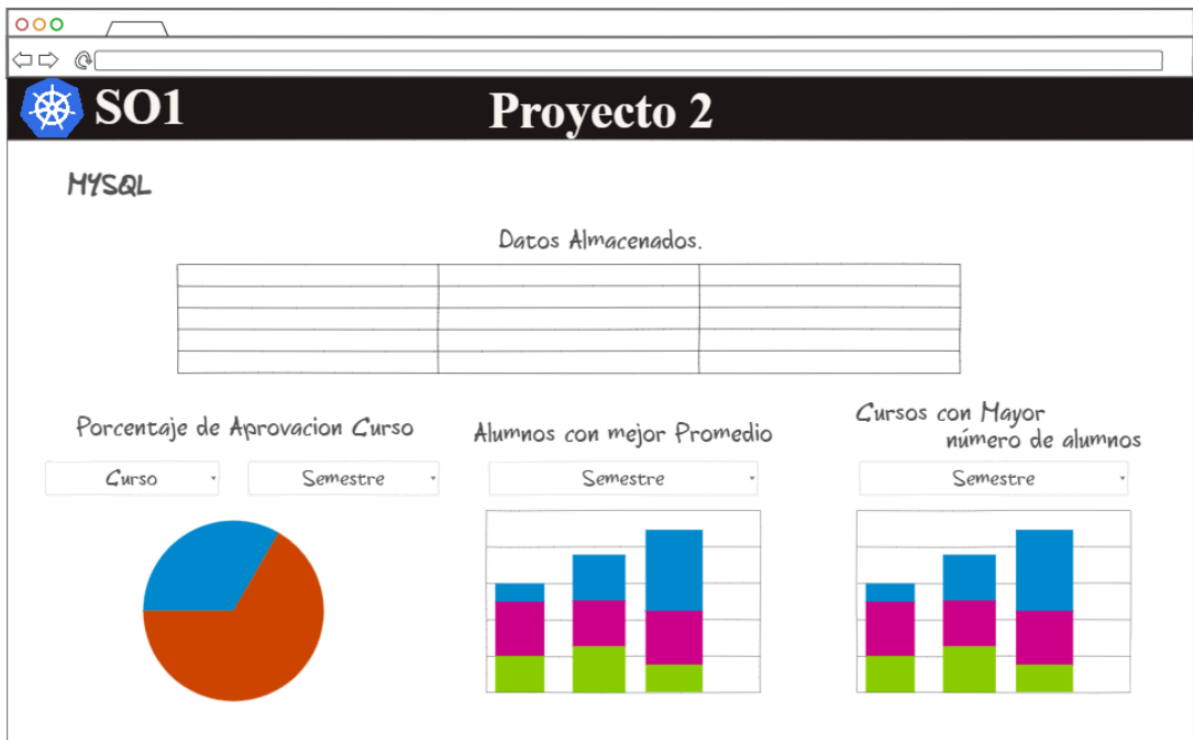
Permite ejecutar aplicaciones en contenedores de manera completamente gestionada y sin preocuparse por la infraestructura subyacente. Se utilizará el Servicio de Cloud Run para desplegar y virtualizar nuestra aplicación frontend.

## APLICACIÓN WEB

Se debe crear un sitio web para mostrar en tiempo real los datos cargados, React para su desarrollo, este debe ser un Dashboard y mostrar lo siguiente:

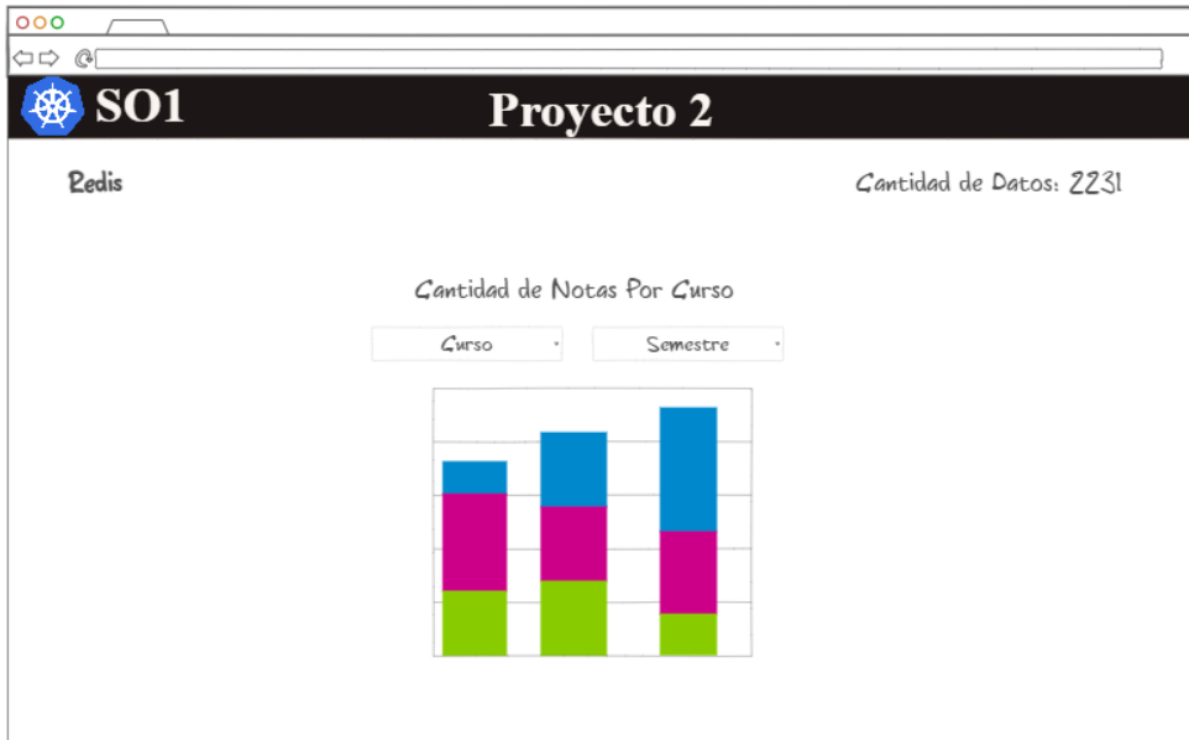
MYSQL (Parte estática)

- Datos Almacenados.
- Gráfica Circular de las Notas de un Curso en un semestre. (No. Aprobados y Reprobados).
- Gráfica de Barras de Cursos con Mayor número de alumnos en un semestre específico. (Mostrar Top 3).
- Gráfica de Barras de Alumnos con mejor Promedio (Mostrar únicamente un Top 5).



## REDIS (Parte dinámica)

- Cantidad Total de Registros en Tiempo Real.
- Datos Almacenados.
- Cantidad de Notas en un Curso y Semestre Específico.



## Github

- El código fuente debe de ser gestionado por un repositorio privado de Github
- Nombre de la carpeta en el repositorio de tareas: **proyecto2**
- Agregar a los auxiliares al repositorio de GitHub: **DannyLaine158** y **JhonathanTocay2020**

## Calificación

- Al momento de la calificación se verificará la última versión publicada en el repositorio de GitHub
- Cualquier copia parcial o total tendrán nota de 0 puntos y serán reportadas al catedrático y a la Escuela de Ciencias y Sistemas.

## Entregables

- **La entrega se debe realizar antes de las 23:59 del 31 de octubre de 2023.**
- **La forma de entrega es mediante UEDI subiendo el enlace del repositorio.**