

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

SISTEMAS OPERATIVOS 1 SECCIÓN N

ING. SERGIO ARNALDO MENDEZ AGUILAR

AUX. GERMAN JOSÉ PAZ CORDÓN

SEGUNDO SEMESTRE 2022



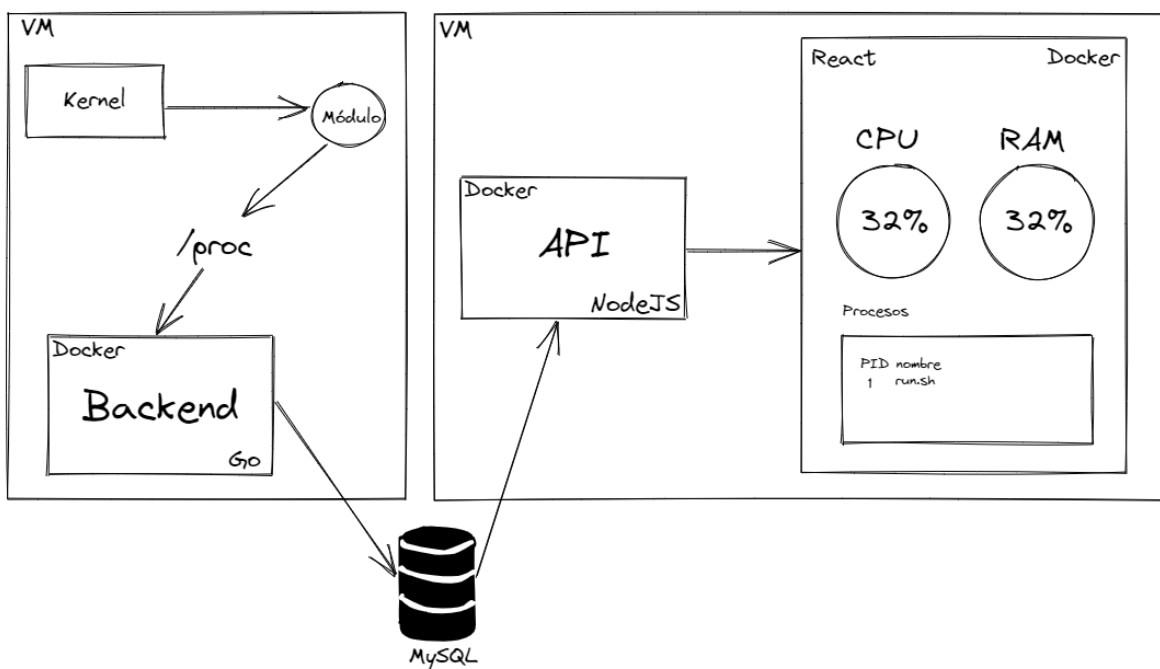
# PRÁCTICA 2

## GCP, MONITOREO DE RECURSOS Y MÓDULO DE KERNEL

### OBJETIVOS

- Conocer el kernel de Linux y sus módulos.
- Comprender y utilizar herramientas de Google Cloud Platform.
- Utilizar máquinas virtuales de GCP.
- Creación de módulos.
- Utilizar CloudSQL.

### ARQUITECTURA



## DESCRIPCIÓN

Se deberá implementar un dashboard simple para obtener información sobre la memoria RAM y CPU del sistema haciendo uso de módulos del kernel escritos en C para obtener la información y escribirla en archivos dentro de la carpeta / para posteriormente leerlos con Golang. Estos datos leídos deberán ser almacenados en una base de datos MySQL para luego ser obtenidos por una API desarrollada en NodeJS. El estudiante deberá utilizar React para realizar la interfaz gráfica que permita ser visualizada a través del navegador. Por último, se le solicita hacer uso de máquinas virtuales de Google Cloud, así como también utilizar CloudSQL para la base de datos.

## MÓDULOS POR IMPLEMENTAR

### Módulo de Memoria RAM (sysinfo)

El módulo deberá sobrescribir un archivo en el directorio **/proc**.

#### Características por implementar:

- Importar librería **<sys/sysinfo.h>**
- Debe imprimir el número de carnet al cargar el módulo (**insmod**).
- Debe imprimir el nombre del curso al descargar el módulo (**rmmod**).
- La información que se mostrará en el módulo debe ser obtenida por medio de los struct de información del sistema operativo y no de la lectura de otro archivo.
- El nombre del módulo será: ram\_<carnet>>

### Módulo CPU (task\_struct)

El módulo deberá sobrescribir un archivo en el directorio **/proc**.

#### Características por implementar:

- Importar librerías: **<linux/sched.h>**, **<linux/sched/signal.h>**
- Debe imprimir el nombre del estudiante al cargar el módulo (**insmod**).
- Debe imprimir "Segundo Semestre 2022" al descargar el módulo (**rmmod**).
- La información por mostrar debe ser obtenida por medio de los struct de datos del sistema operativo y no de la lectura de archivos o comandos de consola.
- El nombre del módulo será: cpu\_<carnet>>

## APLICACIÓN WEB

La aplicación web permite visualizar gráficas dinámicas que muestren:

- Uso del CPU
- Uso de la memoria RAM del servidor.
- La aplicación web permite mostrar la información básica de los procesos que se ejecutan y de sus hijos si tuviesen.

### **Página Principal**

Esta debe mostrar de manera tabulada todos los procesos que están siendo ejecutados en el servidor, así como un resumen general de los procesos.

La información general que se debe mostrar es la siguiente:

Procesos en ejecución	Número de procesos en ejecución (running)
Procesos suspendidos	Número de procesos suspendidos (sleeping)
Procesos detenidos	Número de procesos detenidos (stopped)
Procesos zombie	Número de procesos zombie
Total de procesos	Número total de procesos

Además, se debe mostrar de manera tabulada la siguiente información de cada proceso:

PID	Identificador del proceso
Nombre	Nombre del proceso
Usuario	Usuario que ejecutó el proceso
Estado	Estado en el que se encuentra el proceso
%RAM	Porcentaje de utilización de RAM por el proceso

En esta sección se deberá mostrar un árbol de procesos dinámico, es decir, se debe mostrar una lista de procesos que permita seleccionar uno y desplegar a todos sus hijos con PID y nombre.

### Monitor de CPU

El monitor de CPU debe mostrar la información del consumo de CPU del servidor, en el cual se podrá visualizar la siguiente información:

- Porcentaje de CPU utilizado

### Monitor de RAM

El monitor de memoria RAM es similar al de CPU, debe mostrar la información del consumo de RAM del servidor, en el cual se podrá visualizar la siguiente información:

- Porcentaje de consumo de RAM

## Máquinas Virtuales

Deberá implementar dos máquinas virtuales de Google Cloud. Una de ellas alojará el módulo de kernel y el backend necesario para almacenarlo en la base de datos. Mientras que la segunda se utilizará para alojar la API y su respectivo frontend.

## RESTRICCIONES

- La práctica se realizará de manera individual.
- La aplicación debe tener un aspecto profesional.
- La interfaz gráfica debe ser realizada con React.
- El servidor debe ser realizado con Golang.
- La API debe ser realizada con NodeJS.
- La obtención de la información se hará a través del directorio /proc.
- Cualquier copia parcial o total será reportada a la Escuela de Ciencias y Sistemas para que proceda como indica el reglamento.
- Las máquinas virtuales deberán ser creadas utilizando GCP.
- Utilizar un repositorio de GitHub, el cual debe ser privado con el nombre: **so1\_practica2\_<<carnet>>**.
- El código fuente debe ser administrado por medio de un repositorio de github, al momento de la calificación se bajará la última versión.
- Agregar al auxiliar al repositorio: GermanJosePazCordon

## ENTREGABLES

- Enlace del repositorio de GitHub.
- Manual técnico con explicación de todos los componentes utilizados en la práctica. Este manual debe ser un archivo .md (Markdown), por lo que deberá estar en el README del repositorio.

## FORMA DE ENTREGA

- Mediante UEDI y Classroom, subiendo el enlace del repositorio.

**La entrega se debe realizar antes de las 23:59 del 11 de septiembre de 2022.**