

PROYECTO FINAL

Elaborado por: M. en C. Ukranio Coronilla

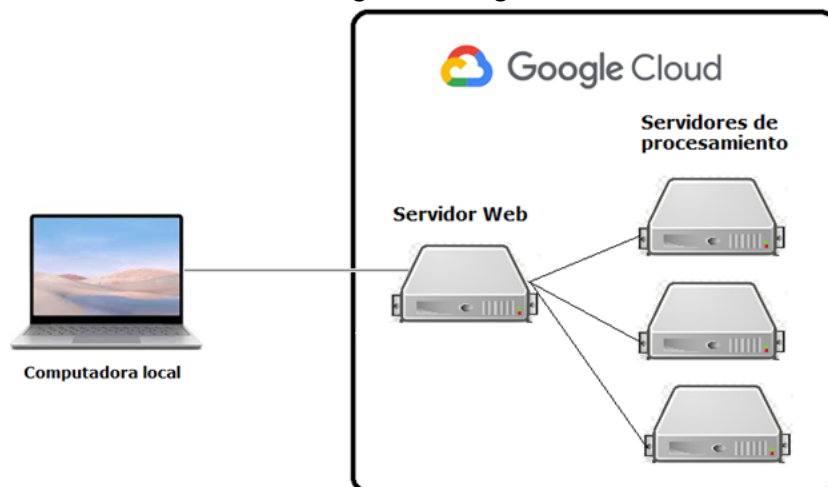
El presente proyecto equivale a dos proyectos (5 y 6) por su complejidad, y debe programarse en lenguaje Java además de utilizar el protocolo HTTP para la comunicación.

En este caso se programará un sistema distribuido en la nube que permite realizar consultas sobre un conjunto de textos almacenados en hasta tres servidores. En todos los casos debe haber una distribución de la carga de trabajo de manera que todos los servidores se encuentren ocupados durante la consulta en todo momento.

Los textos donde se realizarán las búsquedas se podrán descargar desde la siguiente liga:

https://drive.google.com/file/d/1BEdeO-yzpTQcbm7_t1am7QKVkdYWyTXD/view?usp=sharing

La arquitectura del sistema se muestra en la siguiente imagen:



La consulta que podrá hacerse desde la interfaz del navegador web es la siguiente:

El usuario introduce el número de palabras n en el navegador web y este deberá devolver el número de servidores activos (1 a 3) que realizaron el procesamiento, el tiempo que demoró el procesamiento de la consulta en el formato segundos y milisegundos, la lista de enunciados idénticos que se encuentran en cada par de textos y que se conforman de n palabras (no considerar mayúsculas o minúsculas y quitar todos los signos de puntuación). La lista de enunciados debe aparecer ordenada alfabéticamente de la a hasta la z.

Por ejemplo, si el usuario introduce un valor de 3 podría aparecer como resultado la siguiente respuesta:

```
Numero de servidores activos: 2
Tiempo de procesamiento: 2 segundos con 345 milisegundos
```

```
"el sol brillante" aparece en "MUJERCITAS" y en "LA CASA DE LOS ESPIRITUS"
"lo miraba feliz" aparece en "ROBIN HOOD" y en "YO ROBOT"
"pasan los días" aparece en "YO ROBOT" y en "LASARILLO DE TORMES"
"una señora alta" aparece en "ROBIN HOOD" y en "YO ROBOT"
```

...

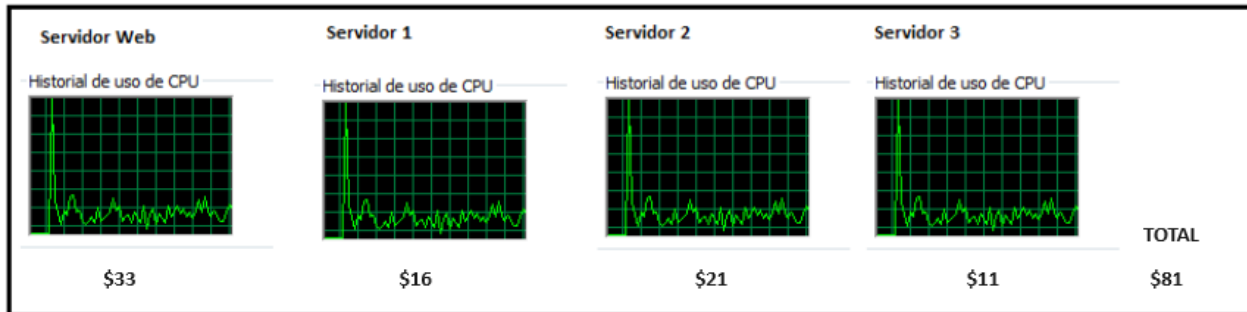
Observe que si se tienen n textos entonces se tendrán que hacer $n!/(2*(n-2)!)$ comparaciones.

De manera que cada una de esas frases se componen de tres palabras y se pueden encontrar exactamente en cada par de textos.

La distribución del trabajo debe realizarse de manera equitativa de tal manera que los servidores de procesamiento se encuentren trabajando durante el periodo que dure la consulta, y para verificar que esto sucede así tendremos un sistema de monitoreo.

El sistema de monitoreo debe tener una interfaz gráfica (en java o en un cliente web) para que un usuario que sea el administrador del sistema distribuido pueda visualizar el historial del porcentaje de uso de CPU para cada una de las cuatro instancias en la nube (servidor web y los cuatro servidores de procesamiento) de manera simultánea.

Asimismo, se debe visualizar un costo por el cómputo realizado, el cual que se calcula con el área bajo la curva del historial de uso de cada CPU multiplicado por un factor f_1 (propuesto por el programador):



Tanto las gráficas como los costos (individual y total) se tendrán que actualizar cada segundo.

El sistema debe ser tolerante a fallos de tal manera que si uno de los servidores se cae (o lo matamos) a mitad de una consulta, el sistema deberá entregar de todos modos la respuesta correcta.

Por otra parte, el sistema deberá ser auto recuperable en el caso de que un servidor de procesamiento caiga (lo matamos sin que se esté haciendo una consulta) deberá levantarse automáticamente un nuevo servidor, de manera que siempre estén los 3 servidores de procesamiento activos. Para cumplir estos últimos requerimientos necesitamos utilizar la herramienta Apache ZooKeeper que revisaremos en clase durante las siguientes semanas, pero todo lo demás se puede programar con los conocimientos adquiridos hasta ahora.

Para crear el video demostrativo de su proyecto deberá mostrar y en este orden:

1. El servidor web y los tres servidores de procesamiento ejecutándose en la nube y en cuatro instancias. Si se ejecutan de manera local se le restarán 3 puntos a su calificación final.
2. El navegador web en pantalla completa y la consulta con $n = 4$ y 3 servidores activos, así como su respectiva respuesta (sólo deben aparecer los títulos de los libros en mayúsculas). (Valor 2 puntos)
3. El navegador web en pantalla completa y la consulta con $n = 4$, y 2 servidores activos (debe verse como mata a uno de los servidores de procesamiento antes de iniciar la nueva consulta) así

como su respectiva respuesta la cual debe ser idéntica al inciso anterior y ocupar mayor tiempo (usar dos pestañas del navegador, una para el inciso 2 y otra para el 3). (Valor 3 puntos)

4. Abrir en otra terminal o en otra pestaña del navegador el sistema de monitoreo gráfico donde se observe que no se están usando recursos de CPU, y ejecutar una consulta con $n = 3$ y 3 servidores activos para observar que se utilizan todos los recursos todo el tiempo durante el procesamiento de la consulta. (Valor 2 puntos)
5. Mostrar que el sistema es tolerante a fallos al matar a uno de los servidores a mitad de una consulta repitiendo con $n = 3$ y pasando de 3 servidores activos a 2 servidores activos, mostrando que la respuesta es igual a la del inciso anterior. (valor 1.5 puntos)
6. Mostrar la versión autorecuperable donde se hace una consulta con $n = 5$ y tres servidores activos, posteriormente se mata uno de los servidores de procesamiento lo cual debe reflejarse en el sistema de monitoreo, también debe reflejarse que se levanta automáticamente otro servidor de procesamiento y posteriormente se vuelve a hacer una consulta con $n = 5$. (valor 1.5 puntos)

Importante: Este proyecto es individual. Suba el código fuente de cada clase en un archivo separado, y cada archivo de código que suba debe contener al inicio como comentario el número de proyecto, su nombre completo y el grupo al que pertenece, de no hacerlo así se le descontará un punto de la calificación. No suba archivos class, tampoco los textos, archivos comprimidos ni ligas a sitios web externos pues no le será tomado en cuenta el proyecto. Asimismo, deberá subir un video breve mostrando con claridad los seis incisos que se indican en el orden indicado. Se recomienda utilizar OBS Studio con baja resolución y no es necesario que hable en el video.

Cualquier duda del proyecto favor de hacérmela saber en el horario de clase.