

Universidad de los Andes - ISIS 2503 Arquitectura y diseño de software-201720
Grupo 4 - Entrega parcial Experimento 2

Escenarios de prueba:

Para realizar las pruebas de las diferentes alertas se implementaron ciertos escenarios de pruebas según la alerta que se quiere probar.

Fuera de rango:

Mediante el sistema integrador IoT, basado en Node-red, se envían datos de manera que se generen las alertas de fuera de rango. Estos datos son simulados, dado que no se cuentan con los sensores reales. Los flujos de la implementación presentada se encuentran en la carpeta "Nodered".

La función utilizada para esto se muestra a continuación:

```
Function
1 var res = {};
2
3 //Temperatura
4 temp = 0;
5 var str = temp + "\t" + "C" ;
6 //Ruido
7 noice = 100;
8 str = str + "\t" + noice + "\t" + "db";
9 //Monoxido de carbono
10 carbon = 110;
11 str = str + "\t" + carbon + "\t" + "ppm";
12 //Iluminacion
13 illumination = 80;
14 str = str + "\t" + illumination + "\t" + "lux";
15 //Numero de microcontrolador
16 str = str + "\t" + "Microcontrolador Num:" + "\t" + 1;
17
18 res.payload = str;
19 return res;
```

Imagen 1. Función creación datos de prueba alerta fuera de rango.

Fuera de línea:

Se genera un timestamp cada vez que se genera una medición por parte del sensor. Luego de esto, se genera una comparación del último timestamp con la hora actual, si ha pasado cierto tiempo sin recibir información se genera la alerta de sensor fuera de línea.

Actuador ineficiente:

Cuando un sensor se encuentra en fuera de rango, el actuador se activa. Después de una hora de activado el actuador se vuelve a verificar el rango. Si el rango está dentro del rango no se genera la alerta. En caso contrario se genera la alerta de actuador ineficiente.

Diagrama de despliegue:

El siguiente diagrama muestra el diseño que se realizó basado en el análisis de los atributos de calidad deseados. Para poder cumplir con estos, se tomaron decisiones de arquitectura, específicamente tácticas de desempeño. La táctica central del diseño es *mantener múltiples copias de computaciones*. Como se observa en el diagrama a continuación se hizo uso de un balanceador de carga para asignar las tareas entre tres distintos servidores, cada uno en máquinas distintas. El balanceador asigna las tareas basado en el criterio del servidor con menos conexiones activas.

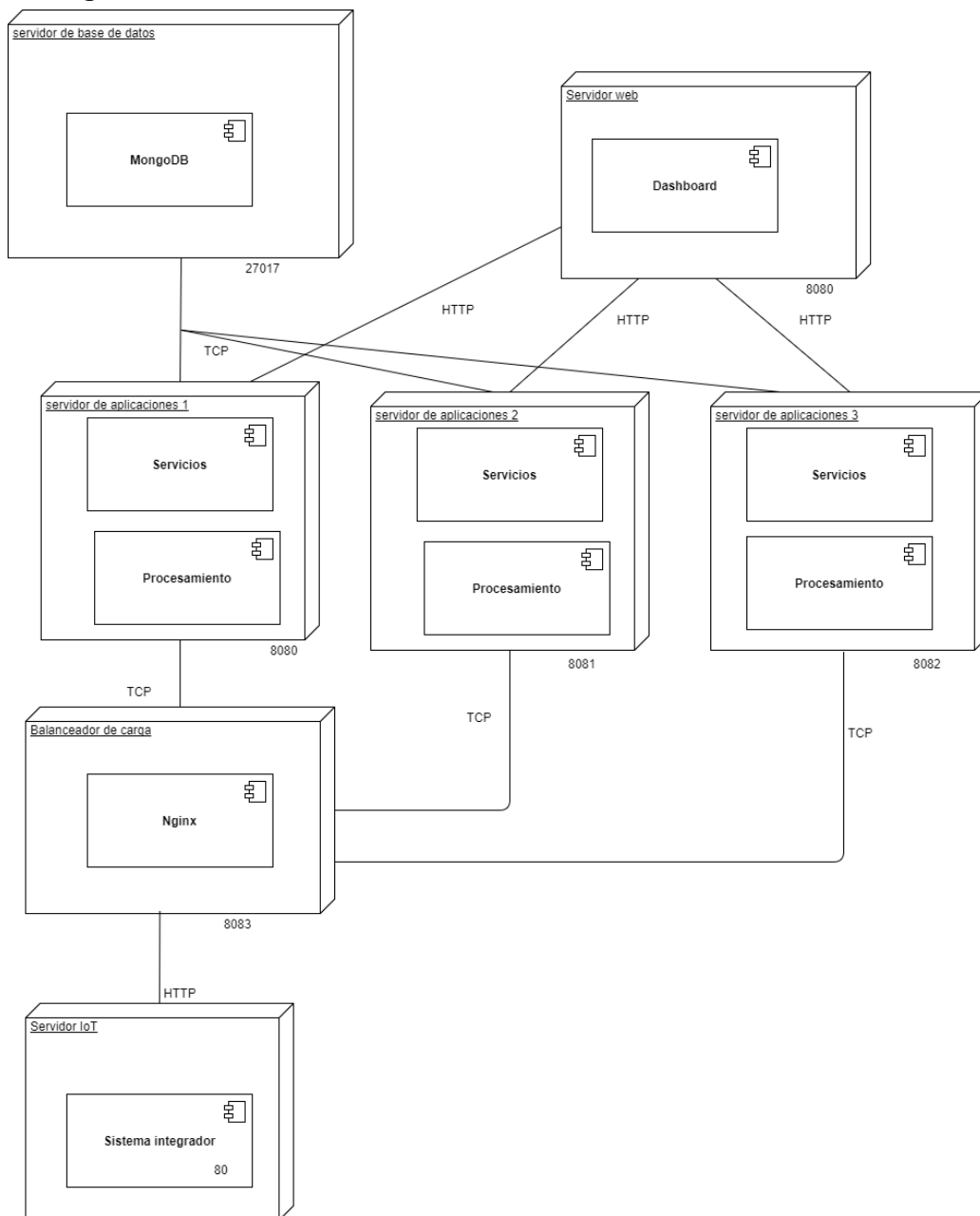


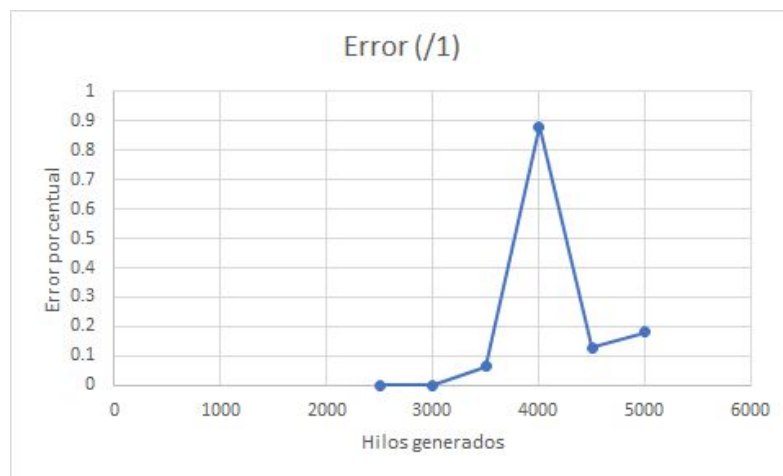
Imagen 2. Diagrama de despliegue.

La capacidad de los servidores de las aplicaciones 1, 2 y 3 y del balanceador de carga, corresponden a la capacidad de las máquinas virtuales en la que están desplegados. Es decir, 12 GB de RAM y dos procesadores de 2.6 GHz. Por su parte, el servidor de base de datos está desplegado en un servidor Mlab remoto y cuenta con 0.5 GB de almacenamiento.

Pruebas de carga:

Para realizar la prueba de carga se hizo uso de JMeter. En coherencia con los requerimientos de calidad propuestos por el enunciado se realizaron pruebas variando la cantidad de hilos generados para realizar solicitudes a los servicios de la aplicación junto con la ventana de tiempo en la que las peticiones son realizadas. A continuación se muestran los resultados.

Ramp up de 1 segundo:



Posterior a esta prueba se procedió a disminuir el ramp up mientras se mantenía la cantidad de hilos generados constante (2500). En cada iteración se disminuía en 0.1 el valor del ramp up, se hacían 3 repeticiones y se obtenía el promedio. A continuación los resultados.

Nótese que no en todos los casos se cumplieron los atributos de calidad. El grupo atribuye esto a la capacidad limitada de