



Tarea 1

Sistemas Distribuídos

SEBASTIÁN I. ALVARADO A.

FELIPE I. GONZÁLEZ G.

Actividad 1

Problema

- ▶ Hacer una aplicación cliente-servidor.
- ▶ La aplicación debe ser ejecutada con “Dockers”.
- ▶ El Cliente debe enviar un mensaje que será guardado en el Servidor.
- ▶ El Servidor debe enviar una respuesta ante el mensaje enviado por el Cliente.
- ▶ Todos los mensajes deben ser guardados en los respectivos archivos “.txt”.

Actividad 1

Solución

- ▶ Se crearon 2 archivos, uno por cada parte de la aplicación (cliente servidor)
- ▶ El archivo cliente y el archivo servidor cumplen con lo indicado en el problema utilizando Python
- ▶ El cliente manda un mensaje “hello” y el servidor le responde “received message”.
- ▶ Para la implementación de Dockers, se creó un “Dockerfile” por cada archivo de código, y un solo archivo “Docker-compose.yml” para poder realizar la ejecución de la aplicación y ver la interacción.

Actividad 2

Problema

- ▶ Hacer una aplicación que simule la arquitectura “Hadoop”.
- ▶ La aplicación debe ser ejecutada con “Dockers”.
- ▶ El Cliente debe enviar un mensaje que será guardado en el Servidor.
- ▶ El mensaje recibido por el Servidor se guardará en uno de los datanodes.
- ▶ El Servidor debe enviar una respuesta ante el mensaje enviado por el Cliente, el cual será el id del “datanode” en donde se encuentra guardado el mensaje.
- ▶ Se deben mantener en ejecución 3 “datanodes”.
- ▶ Todos los mensajes deben ser guardados en los respectivos archivos “.txt”.

Actividad 2

Solución

- ▶ Se crearon 3 archivos, uno por cada parte de la aplicación: cliente, headnode y datanode.
- ▶ El archivo cliente y los archivos del servidor cumplen con lo indicado en el problema utilizando Python.
- ▶ Para la relación entre los nodos, se creó una clase “datanode” en el “headnode” que sirve para poder comunicarse.
- ▶ Para el “heartbeat” se utilizan los mensajes “reponse n” por parte del “headnode” para preguntar, y los mensajes “dokidoki” por parte del “datanode” para responder.

Actividad 2

Solución

- ▶ El cliente envía mensajes mediante un archivo de texto en el cual están escritas las cosas que se le enviarán al servidor (headnode).
- ▶ La ejecución comienza a finalizarse una vez el mensaje “chao” es recibido por el servidor, re-enviando los mensajes “kill” según corresponda.
- ▶ Para la implementación de Dockers, se creó un “Dockerfile” por cada archivo de código, y un solo archivo “Docker-compose.yml” para poder realizar la ejecución de la aplicación y ver la interacción.