

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA



LABORATORIO N°2
SISTEMAS DISTRIBUIDOS
Java WebServices RESTful

Profesor: Nestor Gonzalez nestor.gonzalez.v@usach.cl
Ayudante: Víctor Flores S. victor.floress@usach.cl
Fecha de entrega: Miercoles 11 de diciembre de 2013

OBJETIVO:

Realizar un sitio web de almacenamiento de archivos basado en un sistema distribuido que utilice WebServices basados en RESTful para poder distribuir la carga del sistema.

ENUNCIADO:

Los sitios web en la actualidad no están basados en un único servidor que atiende todas las peticiones de los usuarios, la cantidad de usuarios en la actualidad y la necesidad de hacer que un sistema sea escalable hace necesario de que las distintas funcionalidades del sistema se encuentren distribuidas en distintos servidores.

Su trabajo es desarrollar un sitio web que permita al usuario almacenar archivos en la *nube*, para esto el sitio web debe estar desarrollado en HTML y usar java JSP como lenguaje server-side, la arquitectura del sistema debe ser utilizando un servidor principal que muestra el sitio web y permite el acceso a los usuarios, creación de cuentas, etc. Este servidor principal contiene una base de datos con información de los usuarios registrados, de esta forma, cada vez que el usuario desea listar los archivos que posee almacenados, el servidor principal realiza una consulta mediante RESTful a todas las demás máquinas preguntando por los archivos del usuario que se desea, las máquinas responden a la petición y el servidor principal genera la página HTML que es mostrada al usuario con todos sus archivos. Luego, si el usuario desea descargar un archivo, entonces se le hace la petición al servidor que posee físicamente

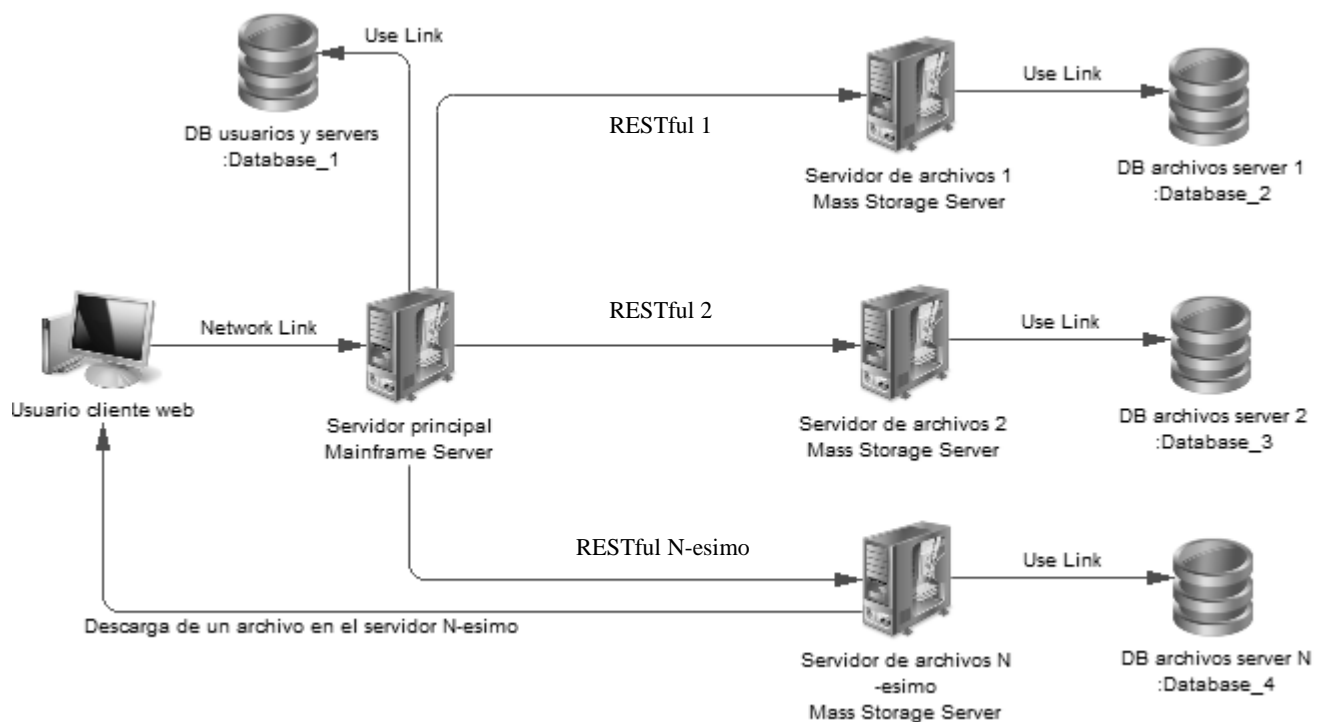
los archivos sin pasar los datos por el servidor principal.

El sistema debe tener los siguientes requisitos:

- Se debe permitir el registro de usuarios en el sistema, comprobando que no se repita el nombre de usuario ni el correo electrónico, que la contraseña no tenga menos de 5 caracteres.
- Se debe permitir el acceso a usuarios registrados y en primera instancia mostrar el listado de los archivos del usuario, sin embargo, los hipervínculos del listado de archivos deben apuntar a los servidores que físicamente poseen el archivo (links REST apuntando a los servidores de archivos), así se evita que los archivos pasen a través del servidor central al momento de ser descargados.
- El servidor principal debe poseer una base de datos relacional, esta base de datos sólo contiene información de los usuarios registrados en el sistema e información de la dirección y puerto por el cual conectarse a los servidores que almacenan los archivos.
- Los servidores que almacenan los archivos deben poseer también una base de datos relacional, esta base de datos sólo contiene información de los archivos que posee almacenados este servidor, como su nombre virtual (El nombre que el usuario le ha dado), su nombre en el sistema de archivos físico del servidor y el nombre del usuario al cual pertenece.
- Cada vez que el usuario desea ver sus archivos (listarlos), el servidor principal se conecta a todos los servidores de archivos pidiendo los datos de los archivos que posee el usuario que desea listarlos, entiéndase “datos de los archivos” como su nombre ante el usuario, su tamaño y su link para descarga; con esta información el servidor principal puede construir el listado de archivos para mostrar al usuario en la vista HTML.
- El sistema debe permitir que el usuario suba al sistema los archivos que desee, para esto, la petición de subida de archivo debe realizarse al servidor principal, el cual mediante una llamada RESTful envía el archivo a alguno de los servidores que los almacenan, la decisión de en cual servidor almacenar el archivo debe ser tomada considerando alguna métrica o de forma aleatoria, por ejemplo: cantidad de archivos por servidor, frecuencia de descarga de los archivos, uso de disco de los servidores, etc.
- Debe controlar la mayor cantidad de errores posibles, por ejemplo si algún servidor de archivos no se encuentra en línea, implicaría que no se mostrasen los archivos que posee tal servidor, pero no debe caerse el sistema en general al ocurrir algún problema de ese tipo.
- El sistema debe permitir eliminar los archivos que el usuario tenga almacenados, para esto, la petición de eliminación debe realizarse al servidor principal, el cual mediante RESTful envía la petición de eliminación al servidor que contenga el archivo que se desea eliminar.

- Cada servidor de archivos debe tener ejecutando una aplicación java web, la cual recibe las peticiones desde el servidor principal utilizando RESTful y recibe las peticiones de descarga directamente desde el usuario.
- Los usuarios al hacer click sobre alguno de sus archivos listados (un botón o sobre el nombre del archivo por ejemplo), la vista HTML ya contiene el link que apunta a una petición REST de alguno de los servidores de archivos, por lo tanto la descarga no pasa a través del servidor principal.

A continuación se presenta un diagrama con la arquitectura del sistema.



Tenga en consideración los siguientes puntos para el desarrollo del sistema:

- Puede utilizar cualquier producto de base de datos.
- Las aplicaciones java web se despliegan en un servidor de aplicaciones, puede escoger entre Glassfish, Apache Tomcat o Jboss.
- La aplicación web del servidor principal puede desarrollarla utilizando java JSF o Spring como frameworks o cualquier framework java.
- Sin ningún problema es posible tener la aplicación web del servidor principal y un servidor de

archivos en un mismo servidor físico. Para tener más de 1 servidor de archivos, es necesario tener más de un servidor físico, a no ser que inicie distintas instancias de un servidor de aplicaciones o utilice distintos nombres para cada despliegue (debe investigar).

- Todo el sistema puede ser desarrollado tanto sobre el sistema operativo MS Windows como también en GNU/Linux, queda a su elección.
- Todos los grupos tienen un plazo de 3 semanas para entregar este laboratorio.
- Cada día de atraso en la entrega del laboratorio implicará 1 punto menos en la nota final.
- La entrega del laboratorio será mediante una subida de archivos a usachvirtual y luego se revisará el laboratorio presencialmente en horario de ayudantía u otro horario previamente acordado.
- El trabajo se realiza en grupos de 2 personas máximo.

Tome en consideración que el sistema a desarrollar no incluye mecanismos de replicación, recuperación de archivos y seguridad avanzada; podría ser más complejo.