

# CLOUD COMPUTING, PRÁCTICA 1: OPEN NÉBULA

Manuel Blanco Rienda

MASTER EN INGENIERÍA INFORMÁTICA [manuelbr@correo.ugr.es](mailto:manuelbr@correo.ugr.es)

# 1. Introducción

El planteamiento de la práctica se basa en la creación de un servicio web, alojado en una máquina en la nube que accede a otra máquina diferente para obtener un servicio de Base de Datos, necesario para desempeñar su función. El servicio web que he implementado para su despliegue en esta arquitectura es un sistema CRUD (Create Read Update and Delete) aplicado a textos que se introducen en la citada Base de Datos.

## 2. Creación de la arquitectura

Lo primero de todo es decidir qué tipo de arquitectura software usaremos para poder poner en funcionamiento mi sistema CRUD. Esta decisión desembocó en la determinación de uso de un servidor apache en una de las máquinas y otro servidor mysql (que hace las veces de servidor de Base de Datos) en la segunda máquina. Una vez decidido esto, ¿sobre qué sistema o sistemas operativos se montará este software? En mi caso: el servidor apache iría sobre una máquina Ubuntu 14.04 y el servidor mysql sobre una Centos 7, de cara a mantener cierta heterogeneidad entre los nodos que forman mi arquitectura. Dado que cada sistema operativo debe ser configurado de forma diferente, distinguimos dos procesos de creación y aprovisionamiento de las mismas.

### 2.1. Creación de la máquina Ubuntu (servidor Apache)

1. A través de una conexión con el sistema Open Nébula instalado en el sistema, procedo a la creación de una plantilla para el sistema operativo Ubuntu 14.04 que está disponible entre las imágenes del repositorio de la nube: `onetemplate create --name "Ubuntu14.04" --cpu 1 --vcpu 1 --memory 1024 --arch x86_64 --disk 10 --nic mcc75575731_vnet --vnc --ssh --net_context`
  - a. El dato de "nic" se ha obtenido con el comando `onevnet list`
  - b. El dato "disk" que hace referencia al id de la imagen del sistema operativo que se usará, se ha obtenido con el comando `oneimage list`.
2. Tras crearse la plantilla, puedo proceder a instanciarla y crear la máquina en sí con el comando: `onetemplate instantiate id`. El parámetro `id` corresponde al número que obtuve cuando creé la template en el paso anterior (también puede verse con el comando `onetemplate list`).

3. Después de ejecutar el comando anterior, la máquina comienza a crearse y podemos ir viendo cómo cambia de estado con el comando: `onevm list`, hasta quedarse en `"runn"`, señal de que la máquina está funcionando correctamente.
4. Ahora es el momento de acceder por ssh a la máquina que he creado para poder instalar y configurar el servidor apache que necesito. Accedo con `"ssh root@ip"`. La ip de la máquina podemos obtenerla con el comando: `"onevm show id_de_la_maquina"`, fijándonos en la línea: `"ETH0_IP"`. Después de aceptar los términos de la conexión por ssh, estaremos dentro de la máquina.
5. A través del comando `"sudo apt-get install apache2"` instalamos el servidor de apache. Lo siguiente es instalar los módulos y plugins de php que necesitará el servidor para poder desempeñar su tarea correctamente. Para ello ejecutamos: `"sudo apt-get install libapache2-mod-php5 php5 php5-mcrypt"`. Por último, tenemos que instalar también el módulo de php que tiene las librerías de conexión con la base de datos de mysql que instalaremos en la otra máquina. Para ello usamos este comando: `"sudo apt-get install php5-mysql"`. Reiniciamos el servicio con `"sudo service apache restart"`.
6. Ya tenemos el servidor funcionando, pero en este momento no está sirviendo nada (tan solo la página de bienvenida de apache en localhost). Dado que la aplicación la he desarrollado primero en local y almacenado en un repositorio de mi cuenta de Github, procedemos a su despliegue en esta máquina para que el proceso de puesta en marcha sea inmediato. Para ello instalamos git con el comando: `"sudo apt-get install git"`. Una vez completado el proceso, ejecutamos: `"git clone https://github.com/manuelbr/CC2\_Master.git"`, que descargará mi repositorio para la asignatura, que tengo en Github. Cuando haya acabado de descargarse, deberemos mover la carpeta `"src"`, dentro de la carpeta práctica 1 al directorio: `/var/www/`, y la renombraremos como `"html"`.
7. Dado que en Ubuntu no hay que configurar un firewall para que permita acceder por el puerto 80 al servidor de forma externa, si en un navegador usamos la dirección: `docker.ugr.es:15065`, podremos ver la pantalla de bienvenida del servicio CRUD (el puerto de la dirección es el que corresponde con la ip de la máquina en la que se aloja el servidor).

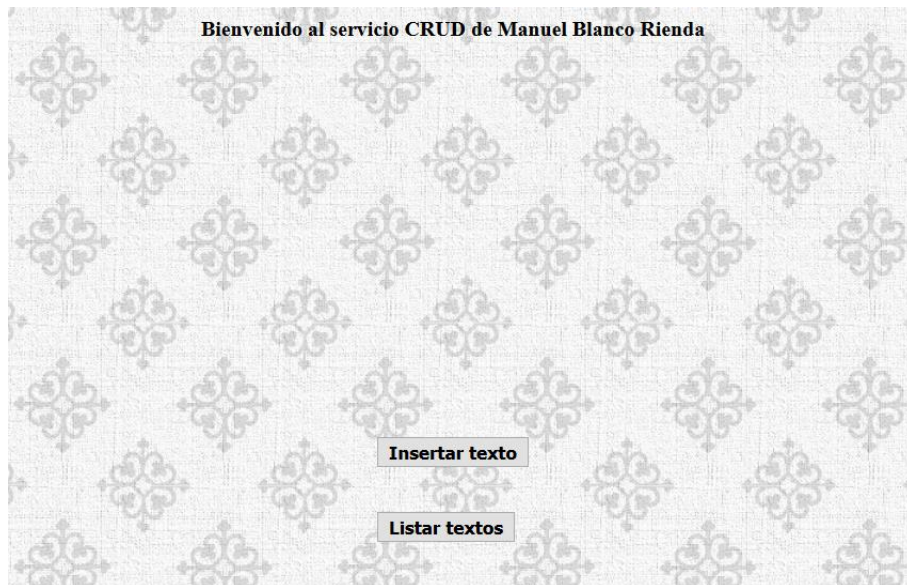


Figura 1: Pantalla de inicio del servicio web

## 2.2. Creación de la máquina Centos 7 (servidor Mysql)

1. Seguimos los mismos pasos (del 1 al 4) que en el anterior apartado, pero especificando diferentes parámetros a la hora de crear la plantilla y la máquina que instancia, correspondientes con la imagen de Centos 7 que queremos instalar (id 9).
2. Tras conectar vía ssh con la máquina Centos, pasamos a instalar el servidor mysql necesario para tener un servicio de Base de Datos con el que tratar desde la aplicación que se encuentra en el apache. Para ello hacemos: “sudo yum update”, para actualizar los repositorios del gestor de paquetes yum de Centos. Tras esto, instalamos el propio servidor de mysql con el comando: “sudo yum install mysql-server” y cuando termine de completarse esta orden, usamos: “sudo /sbin/service mysqld start” para arrancar el servidor.
3. Ahora es el momento de configurar mysql server con la orden “sudo /usr/bin/mysql\_secure\_installation”, en la cual se nos preguntará la contraseña por defecto del usuario root del servidor (ninguna, se deja en blanco) y se establece la nueva contraseña. Después se procede a ajustar algunos temas de privacidad como: el permitir conexiones remotas al servidor (decimos que sí), el reiniciar el servicio iptables tras completar la configuración y demás... A continuación puede verse una captura de pantalla de lo que debe verse en este proceso:

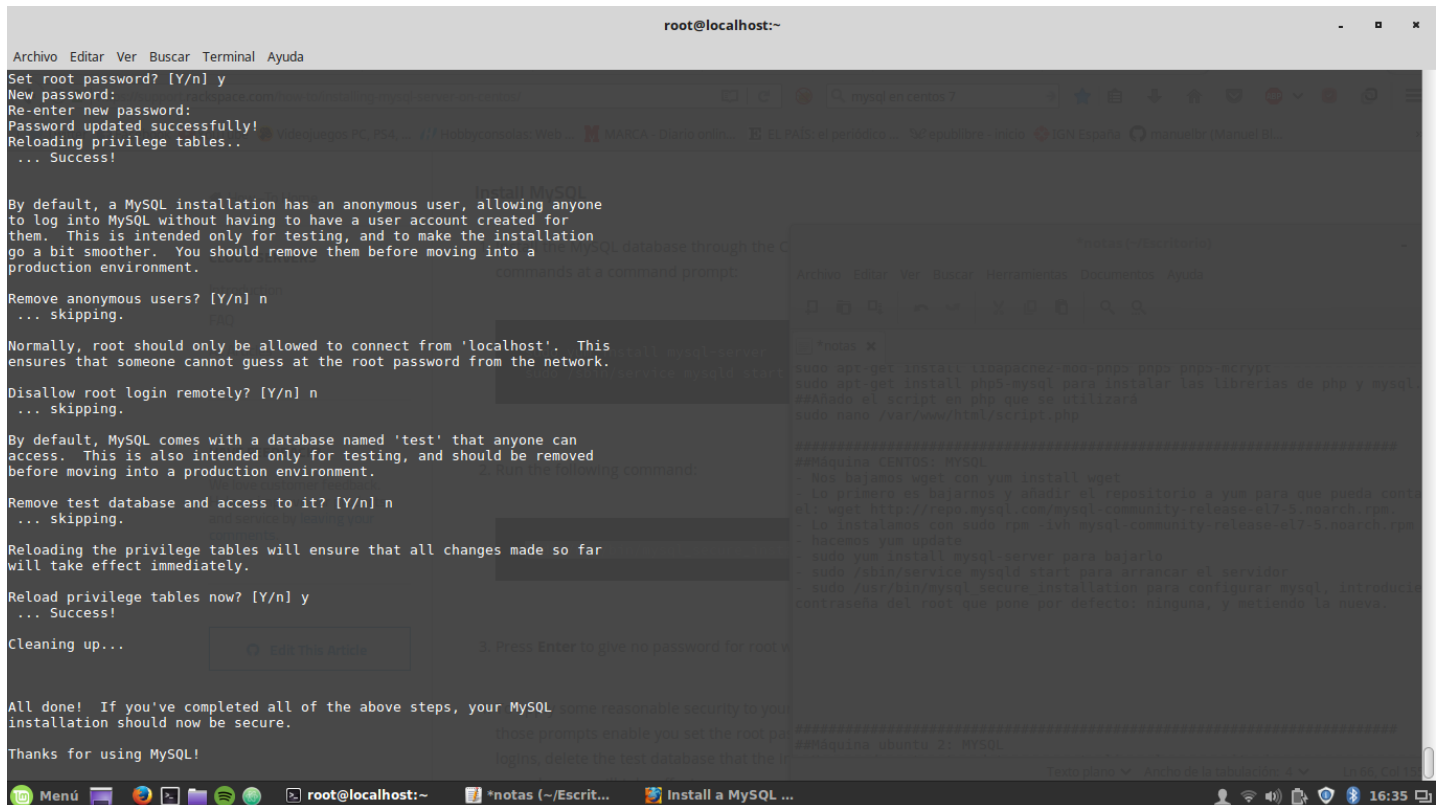


Figura 2: Proceso de configuración de mysql-server

4. Para que el servidor mysql sea accesible desde otra máquina es necesario especificar este detalle al firewall de Centos 7: iptables, con el siguiente comando: “sudo iptables -I INPUT 5 -i eth0 -p tcp --dport 3306 -m state --state NEW,ESTABLISHED -j ACCEPT”, en el que determinamos que queremos permitir la entrada de conexiones al puerto 3306, que es donde espera el servicio mysql. Para aplicar los cambios realizados al firewall ejecutamos: “sudo service iptables restart”.
5. Con el comando “mysql -u root -p” accedemos al servidor mysql vía Shell. Lo primero que hacemos es crear la base de datos que usará mi aplicación: textos, con el siguiente comando: “CREATE DATABASE textos;”. De esta forma, la primera vez que se inserte algo en la base de datos de textos, desde el servicio en apache se creará la tabla “textos” con la que se trabajará. Si ahora comprobamos el funcionamiento del servicio a través de la dirección “docker.ugr.es:15065”, vemos que funciona correctamente.

NOTA: Dado que sólo entramos en la Shell de mysql para crear la base de datos, con un único comando, veo innecesario crear un script para hacer esto.

### 3. Descripción de la aplicación web

Tal y como se ha dicho anteriormente, la aplicación consiste en un servicio de lectura, escritura, actualización y eliminación de cadenas de texto, es decir: un CRUD. Su funcionalidad es la siguiente: cualquier usuario puede acceder a la app a través de la dirección: “docker.ugr.es:15065” y a partir de ahí puede insertar textos o listar los que ya hay en la base de datos. Si accedemos al listado, cada uno de los ítems que lo componen pueden ser actualizados (sustituídos) por un nuevo texto o bien ser eliminados de la base de datos.

En cuanto a la arquitectura software, la aplicación en sí se encuentra alojada en un servidor apache dentro de una máquina virtual Ubuntu 14.04 en Open Nébula. A través de un servidor mysql alojado en una máquina virtual Centos 7, la aplicación obtiene acceso a la base de datos que necesita para funcionar: textos. Es la propia aplicación, una vez conectada a la base de datos la que crea la única tabla que se almacena: textos (se llama igual que la base de datos), la cuál consta de solo dos campos: id (entero, llave primaria de la tabla y auto-incrementable) y texto (tipo text no nulo).

Sobre la arquitectura de la aplicación web, decir que se ha implementado usando php, html para el contenido de la misma y css para determinar su estilo (muy simple y sencillo, dado el objetivo de la práctica). El directorio de archivos fuente se distribuye en un archivo index.php, que conforma la página principal de la web, una carpeta de imágenes (que contiene el fondo de la web), una carpeta php que contiene los scripts en ese lenguaje y otra css, que almacena lo propio al estilo de la aplicación. El flujo que se sigue en el proceso de ejecución de la web es el siguiente:

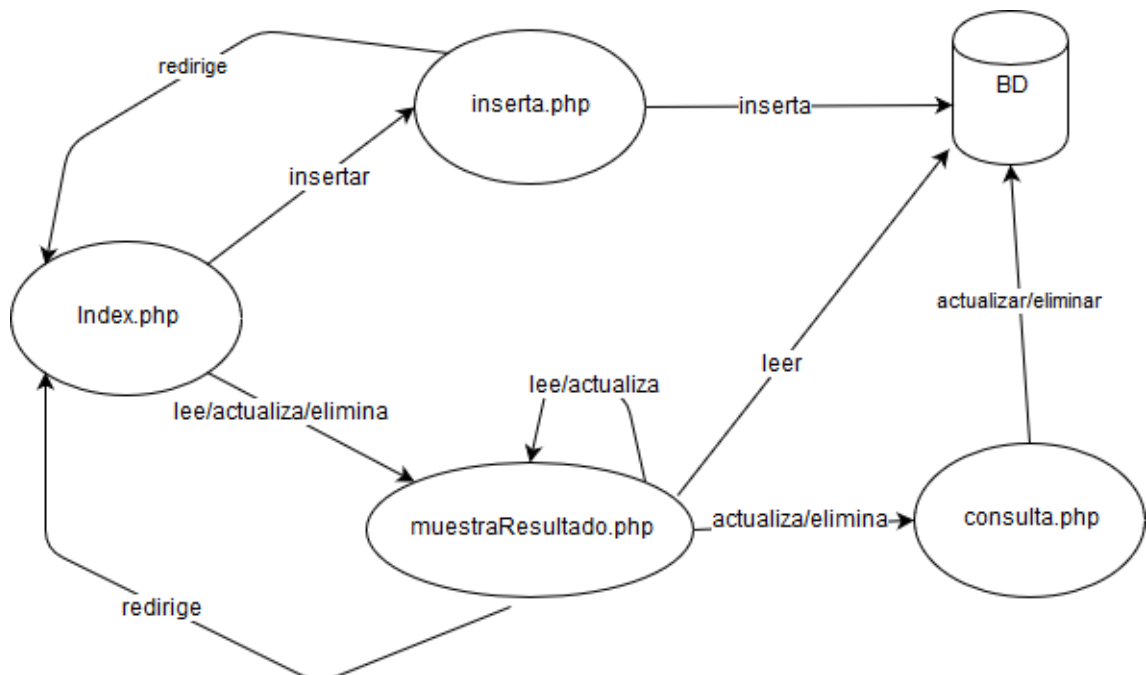


Figura 3: Flujo de procesos entre los fuentes de la aplicación

## 4. Manual de uso de la aplicación

El uso de la aplicación es extremadamente simple, al igual que su funcionalidad. A través de la página principal (<http://docker.ugr.es:15065>) de la web, accedemos a dos opciones: insertar texto (la cual nos lleva a una página en la que podemos escribir un texto en un recuadro y pulsando “insertar texto”, éste se introduce en la base de datos y se nos redirige a la página principal) y listar textos (la cual lista todos los textos que hay almacenados en la base de datos). Si accedemos a esta segunda opción, podremos ver que al lado de cada ítem de la lista aparecen dos opciones: “Actualizar” (que nos lleva a una página parecida a la de inserción, sólo que sustituye el texto que queremos actualizar, por otro) y “Eliminar” (que elimina el texto de la base de datos).

Hay que remarcar el hecho de que dados los problemas que ya se han notificado al profesor de la asignatura con Open Nébulas y los cambios de estado (máquinas apagadas sin motivo aparente, invasión de ips entre compañeros de la asignatura por compartición de red virtual y demás), es posible que a veces no funcione correctamente la aplicación. Sin embargo, hasta el momento al recargar la página se ha conseguido evitar estos problemas.

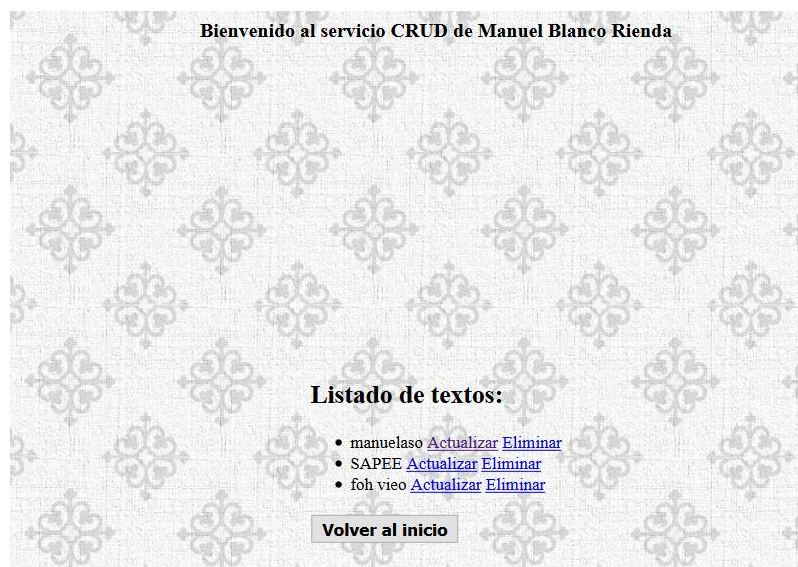


Figura 4: Pantalla de listado de textos



Figura 5: Pantalla de inserción de textos

## 5. Manual de despliegue de la las Máquinas virtuales

Es innecesario repetir otra vez todo el proceso completo, dado que éste ha sido definido como parte de la creación y configuración de las máquinas en los puntos 2.1 y 2.2.