

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



Grado en Ingeniería Biomédica
Arquitectura de Computadores y Sistemas Operativos
Curso 2017-2018

Práctica del segundo parcial
Desarrollo de un escenario de un *Gestor Hospitalaria*

Fecha del documento: 11 May 2018

Índice

	Página
1. Introducción	3
2. Objetivos	3
3. Requisitos	4
4. Recursos proporcionados	5
5. Directrices de desarrollo	5
5.1. Instalación de la aplicación	5
5.2. Instalación del balanceador de carga	5
5.3. Instalación de la base de datos	5
5.4. Acceso a la aplicación	5
6. Partes opcionales	6

1. Introducción

Esta práctica ilustra el desarrollo de la aplicación *Gestión Hospitalaria* con un interfaz web, que se replica en un conjunto de servidores. Esta aplicación es muy sencilla que permite gestionar los pacientes de un hospital, para centrarse en el desarrollo en las operaciones de configuración de la plataforma y el despliegue de la aplicación y los componentes necesarios. El sistema propuesto permite integrar contenidos en varias asignaturas:

- ALED: La aplicación fue desarrollada en esta asignatura.
- BD: En esta asignatura se trabajó con la base de datos MongoDB y se integró la aplicación previa para asegurar la persistencia de los datos de los clientes.
- ARSO: En esta asignatura se han impartido, entre otras cosas, la creación de máquinas virtuales (contenedores), la conexión entre ellos, la creación y configuración de servidores replicados y un balanceador para que los clientes puedan acceder a los servidores con transparencia.

El sistema que se propone ilustra el despliegue y configuración de sus componentes en la computación de la nube. En concreto, esta actividad se basa en servicios como infraestructura (IaaS). Los componentes del sistema se instalarán y se ejecutarán en máquinas virtuales que se ejecutarán sobre la infraestructura creada y disponible en la nube. La infraestructura proporciona al usuario capacidades de procesamiento, almacenamiento, redes y otros componentes. En este sistema habría que realizar como las siguientes:

- Creación de los computadores remotos (posiblemente, serían máquinas virtuales remotas) donde se instalarán, configurar y ejecutar los contenidos del sistema.
- Para cada computador remoto, instalar y configurar el sistema operativo y componentes necesarios, donde destaca el hipervisor para la creación de máquinas virtuales (hipervisor LXD).
- Instalar, configurar y ejecutar los componentes del sistema: servidores, bases de datos, balanceador de carga, etc. En esta actividad, habría que asegurar las comunicaciones entre los componentes. En muchos casos, cada uno de estos componentes ejecutarían sobre diferentes computadores.
- Monitorizar, para asegurar el comportamiento adecuado de los componentes. De esta forma, sería posible crear o borrar componentes para mantener los necesarios para adaptarlos a la demanda de los clientes o si hay fallos.

El sistema en esta práctica tiene un objetivo docente para ilustrar los aspectos más importantes de uno real, acotar el esfuerzo de desarrollo y enfatizar los aspectos de los contenidos de la asignatura. En este caso, todos los componentes se integrarán en contenedores en el mismo computador (potencialmente, en una máquina virtual). El entorno LXD proporciona instrucciones para crear contenedores remotamente, de forma que se podrían ejecutar en otros computadores. De esta forma, el paso de un sistema en un computador a un sistema que se ejecuta en diferentes computadores sería posible con un asequible esfuerzo.

2. Objetivos

El objetivo de esta práctica es desplegar y configurar una aplicación con interfaz web. La práctica final del segundo parcial se consistirá en el desarrollo de un programa en Python (*pfinalp2.py*) que automatice las características que se va a describir en este documento, en el escenario mostrado en la figura 1.

Las características más relevantes del sistema a desarrollar son:

- Aplicación: Disponer una aplicación sencilla de la gestión de hospitalaria *Gestión Hospitalaria*, para gestionar los pacientes de la institución.
- Interfaz web: El acceso a los servicios de la aplicación será por un interfaz web, de forma que se podría acceder desde distintos dispositivos.

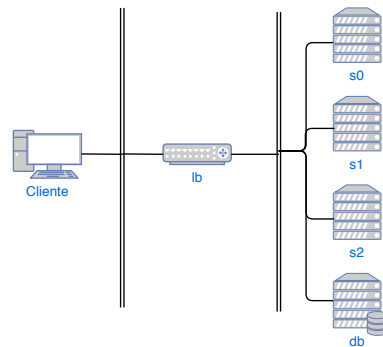


Figura 1: Escenario de la plataforma de un sistema servidor

- **Replicación:** La aplicación se instalará en varios servidores para tolerar fallos y adaptar a la capacidad de acceso.
- **Persistencia:** Los datos de la sistemas y de sus clientes se almacenarán en dispositivos que no sean volátiles.
- **Transparencia:** El acceso a los servicios de la aplicación será con transparencia, si conocer la dirección del servidor concreto al que se va a ejecutar.

3. Requisitos

El programa *pfinal2p1.py* se ejecutará pasándole un parámetro obligatorio que definirá la operación a realizar:

```
$ python3 pfinalp2.py <orden><parámetros>
```

donde el parámetro *orden* puede tomar el valor:

- **configurar**, instalar y configurar los contenedores del sistema, con su requisitos específicos descritos en la sección 5.

Este esquema debe ser compatible con el fichero *pfinalp1.py*, para que sea asequible integrar el conjunto. La operación previa debe proporcionar los siguientes requisitos:

- Desplegar y configurar la aplicación *Gestión Hospitalaria* en un conjunto de servidores, de forma que sean accesibles en el puerto 8080. La aplicación se deberá conectar a una base de datos, para que la información en los servidores sea coherente y con persistencia.
- Instalación y configuración de un balanceador de carga (*haproxy*) que debe recibir peticiones de clientes a la aplicación y distribuirlas entre los servidores para equilibrar las invocaciones.
- Instalar y configurar una base de datos (*MongoDB*), de forma que sea accesible desde los servidores.

4. Recursos proporcionados

Para la realización de la práctica, se proporciona en el *moodle* de la asignatura:

- Una máquina virtual, en la que se han instalado los contenedores que incluyan los servidores, el balanceador de carga y la base de datos. La comunicación está realizada, de acuerdo con la figura previa.
- Una aplicación del gestor para ofrecer los servicios requeridos. La aplicación está configurada para conectarse con una base de datos en un contenedor cuyo nombre es *db*.

5. Directrices de desarrollo

5.1. Instalación de la aplicación

La instalación y configuración de la aplicación se realizará de acuerdo a las instrucciones en la sección 5.2 del laboratorio 7.2.

5.2. Instalación del balanceador de carga

La instalación y configuración de la aplicación se realizará de acuerdo a las instrucciones en el laboratorio 7.1.


5.3. Instalación de la base de datos

La instalación y configuración de la base de datos está basada en la práctica realizada en la asignatura *Base de Datos*, en la que conecta una aplicación de gestión hospitalaria con una base de datos. En este caso, la base de datos está localizada en la misma máquina que la aplicación. En esta práctica, estos componentes están en diferentes computadores. Para ello, se ha modificado la aplicación para que se conecte con la base de datos en una máquina cuyo nombre es *bd*. Este cambio se ha hecho en la aplicación que se proporciona, por lo que no es necesario modificarla.

Las operaciones necesarias son las siguientes:

- Instalación de la aplicación de base de datos *MongoDB*: La forma de instalarlo consiste en ejecutar las siguientes sentencias:

```
 apt-get update
```

```
 apt-get install mongodb
```

- Configuración de *MongoDB*: Es necesario modificar el fichero de configuración */etc/mongodb.cfg*. Para ello, hay que buscar el texto:

```
bind_ip = 127.0.0.1
```

por el texto siguiente. Con ello, se indica a *MongoDB* que reciba y trata operaciones remota a la dirección IP que se proporciona.

```
bind_ip = 127.0.0.1,10.0.0.11
```

El fichero se puede bajar o cargar del contenedor como se ha hecho en otras prácticas. Una forma de modificar el fichero como se puede hacer leyendo línea a línea el fichero y escribiendo en otro, excepto en la línea que empieza *bind_ip*, en la que añada la dirección IP del contenedor.

5.4. Acceso a la aplicación

En esta práctica, no es necesario un contenedor para representar al cliente. En este caso, se debe acceder a la aplicación desde la máquina anfitriona, para simplificar el desarrollo de esta práctica. La instalación

de un navegador gráfico en un contenedor que se comporte como un cliente es más complicado al requerir el entorno gráfico. Una alternativa sería usar el navegador texto *lynx*, aunque es menos visible y fácil de utilizar. Por otro lado, no añade contenidos docentes relevantes en esta práctica.

El acceso a la aplicación puede ser mediante la llamada:

 <http://lb:8080>

6. Partes opcionales

Como funcionalidades adicionales se propone incluir:

- Integración de esta práctica, de forma que se pueda desarrollar el conjunto del sistema.
- Crear dinámicamente servidores con la aplicación. Se recomienda crear una imagen a partir de un contenedor operativo. Cuando se requiera, se creará el contenedor a partir de la imagen y asignar la dirección IP requerida. También sería necesario actualizar el fichero de *haproxy* para incluir el contenedor nuevo.