

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



Grado en Ingeniería Biomédica
Arquitectura de Computadores y Sistemas Operativos
Curso 2024-2025

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

Fecha del documento: 8 abril 2025

Índice

	Página
1. La entrega de la práctica	3
2. Examen de la práctica	3
3. Introducción y objetivos	4
3.1. Introducción	4
3.2. Objetivos	4
4. Requisitos funcionales	5
4.1. Desarrollo del escenario de un servidor web	5
4.2. Despliegue y uso de una base de datos remota	6
5. Directrices con la gestión de un contenedor remoto	7
5.1. Consideraciones para el despliegue remoto	8
6. Partes opcionales adicionales	8
7. Recursos proporcionales e información adicional	9
8. Normas	9

1. La entrega de la práctica

Por favor, usad las siguientes directrices. En caso contrario, se puede penalizar la calificación:

- La práctica se puede entregar antes de las 23:59 del 22 de mayo.
- El nombre del fichero principal de la práctica tiene que ser `pfinal2.py`.
- Si la práctica tiene varios ficheros, se debe subir un fichero comprimido incluyéndolos.
- Cada grupo tiene que subir un sólo fichero. Sólo un miembro del grupo tiene que subirlo. Los nombres de las personas del grupo se deben incluir al principio del fichero principal, como comentarios.

Todas las instrucciones que hay que usar en esta práctica para la configuración del sistema se han ilustrado y usado en los laboratorios de esta asignatura. Es muy recomendable revisar los enunciados de las actividades realizadas en el desarrollo de esta propuesta.

2. Examen de la práctica

Después del plazo de entrega, se propondrán varias sesiones para la evaluación. Los grupos seleccionarán un horario en una sesión. La evaluación se realizará en el laboratorio B123 y todos los miembros del grupo tendrán que asistir en el evento.

En la evaluación, se comprobará la corrección del programa desarrollado. Los alumnos deberán contestar una serie de preguntas para comprobar que han adquirido las competencias y contenidos asociados a la práctica. La calificación será individual de cada persona, dependiendo de sus respuestas.

En la evaluación de la práctica se valorará:

- Corrección del programa desarrollado, de acuerdo a los requisitos funcionales establecidos.
- Corrección de las respuestas a las preguntas planteadas.
- Uso de *logs*: esta función es muy útil para detectar errores de ejecución del código. En concreto, es más importante en el desarrollo del código y en la depuración. Es aconsejable usar diferentes niveles para seleccionar las trazas requeridas, por ejemplo, cuando se depura o en ejecución.
- Calidad y estilo del código: El código se escribe pocas veces y se lee muchas. Todos los lenguajes de programación disponen de guías de estilo para ayudar a crear código más correcto y legible. <https://pep8.org/> es un enlace para *Python*. Se aconseja su uso.
- Uso de módulos: El uso de paquetes y funciones es básico para disponer un código bien estructurado que facilita su desarrollo y legibilidad. Además, es adecuado estructurar el código y encapsular módulos relacionados en diferentes ficheros.
- Mostrar que las soluciones usadas sean las más adecuadas en los laboratorios de esta asignatura.

La calificación de la práctica se basará en:

- Desarrollo del escenario de un servidor web: 6,5 puntos.
- Despliegue y uso de una base de datos remota (opcional): 2,5 puntos.
- Partes opcionales adicionales: 1 punto ¹

¹Este valor se obtiene haciendo, al menos, una actividad opcional

3. Introducción y objetivos

3.1. Introducción

Esta práctica ilustra el desarrollo de la aplicación *Gestión de un Hospital* con un interfaz web, que se replica en un conjunto de servidores. Esta aplicación permite gestionar los pacientes de un hospital. Se ha decidido usar esta aplicación sencilla para centrarse en el desarrollo de las operaciones de configuración de la plataforma y su despliegue, y los componentes necesarios. El sistema a desarrollar integrará los contenidos de varias asignaturas:

- *Algoritmos y estructuras de datos*: La aplicación que se va a usar es un servidor web similar al desarrollado en esta asignatura y con la misma funcionalidad.
- *Bases de datos*: En esta asignatura se ha desarrollado una aplicación web con la base de datos *MongoDB* para asegurar la persistencia de los datos de los clientes.
- *Arquitectura de computadores y sistemas operativos*: En esta asignatura se han impartido, entre otras cosas, la creación de máquinas virtuales (contenedores), su conexión, la creación y configuración de servidores replicados y un balanceador, para que los clientes puedan acceder a los servidores con la 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 infraestructura como servicio (*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 actividades como las siguientes:

- Creación 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 (*LXD*) para la creación de máquinas virtuales.
- Instalar, configurar y ejecutar los componentes del sistema: servidores, bases de datos, balanceador de carga, etc. En esta actividad, hay que asegurar las comunicaciones entre los componentes. En muchos casos, cada uno de estos elementos se 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.
- Desplegar, ejecutar y acceder remotamente a una base de datos en un contenedor.

El sistema en esta práctica tiene el objetivo docente para ilustrar los aspectos más importante de un caso real, por lo tanto, se va a acotar el esfuerzo de desarrollo y enfatizar los aspectos de los contenidos de la asignatura. En este caso, se parte de integrar los componentes en contenedores en el mismo computador (potencialmente, en una máquina virtual). Se propone una extensión para que la base de datos se despliegue en un computador remoto². 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 esfuerzo asequible.

3.2. Objetivos

El objetivo de esta práctica es desplegar y configurar una aplicación con interfaz web. La práctica final del segundo parcial consistirá en el desarrollo de un programa en Python, que emule un sistema real y

²Revisad las funciones realizadas en el Laboratorio 6.3.

automatice las características que se va a describir en este documento. Además, se propone una actividad opcional para desplegar y arrancar un servidor de una base de datos en un computador remoto.

En la figura 1 se muestra el escenario del objetivo obligatorio.

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

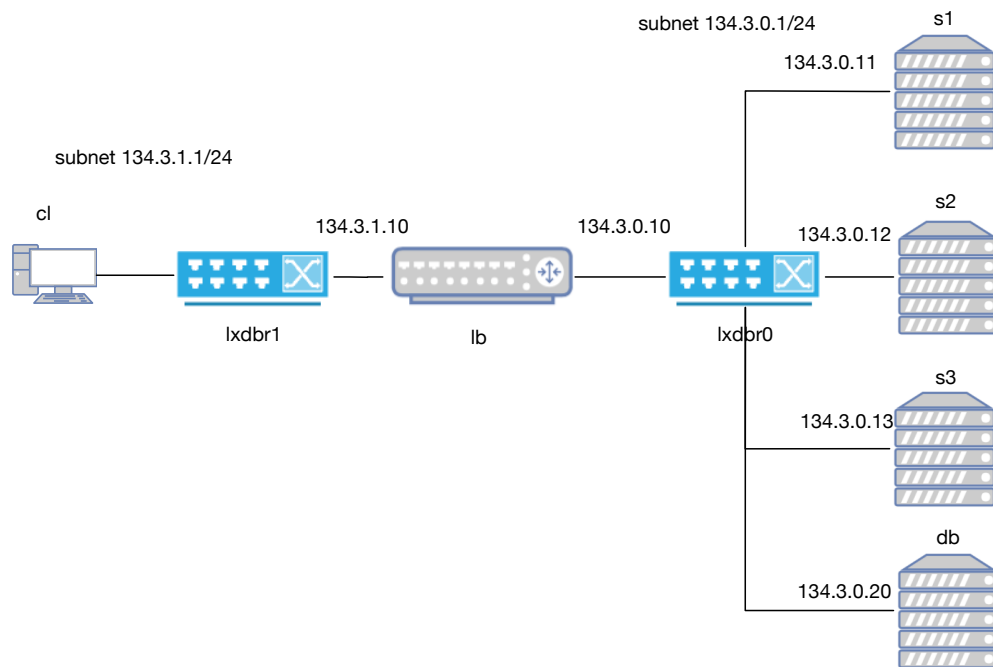


Figura 1: Escenario de la plataforma de un sistema servidor Web

Además, se proponer extender al escenario previo, para que la base de datos se ejecute en un computador diferente y sea accesible remotamente, como se muestra en la figura 2.

4. Requisitos funcionales

4.1. Desarrollo del escenario de un servidor web

En esta práctica se debe desarrollar un sistema que proporcione la aplicación *Gestión Hospitalaria*, hecho en la asignatura *BBDD*. Las funciones que se requieren son:

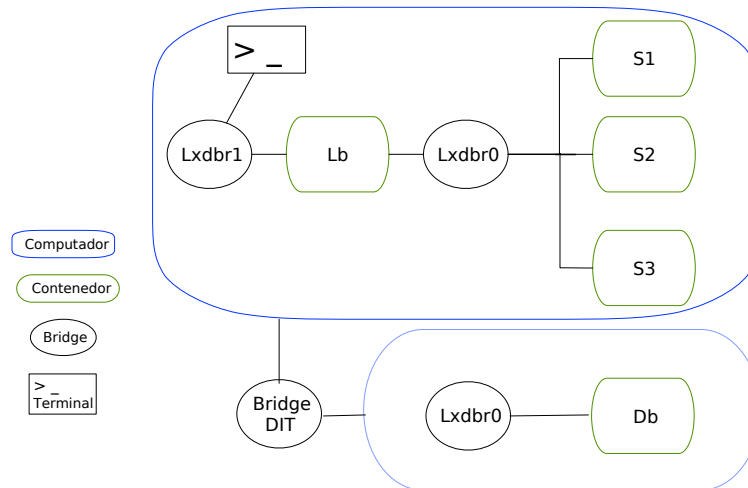


Figura 2: Escenario de la plataforma de un sistema servidor Web con bases de datos remoto

- Crear un conjunto de servidores, que proporcionan los mismos servicios de esta aplicación con un interfaz *Web*. En este caso, se deberá usar el servidor *Node.js* ³.
- Crear un balanceador de carga que recibe peticiones a los mencionados servicios de los clientes y los reparte entre los servidores, según un algoritmo seleccionado.
- Crear un servidor que proporcione un servicio de acceso a una base de datos con *MongoDB*. La base de datos debe gestionar la información de los usuarios de la aplicación.
- Los servidores de la aplicación deben interaccionar con el servidor de la base de datos, para gestionar la información de los pacientes.

Los grupos deben desarrollar el programa *pfinal2.py* que implemente las funciones descritas y que deberá extender el programa desarrollado en la práctica del primer parcial. Por tanto, debe invocarse como la siguiente orden:

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

donde el parámetro *orden* puede tomar los siguientes valores:

- Las mismas órdenes desarrolladas en la práctica mencionada. En caso de que sea necesario, se podría extender algunas de las funciones de éste. En concreto, habrá que integrar la creación y gestión del servidor con la base de datos.
- **configure**: instalar y configurar los contenedores del sistema, según se ha descrito previamente. Aclarar configure. Concretar qué funciones hay que hacer

4.2. Despliegue y uso de una base de datos remota

Esta actividad es opcional. El objetivo es:

- Desplegar y arrancar remotamente una base de datos en un contenedor.
- Configurar el contenedor de la base de datos para que esté accesible remotamente.
- Configurar la aplicación web para acceder remotamente a la base de datos

Los grupos interesados deben ampliar el fichero *pfinalp2.py* con la orden **configure** que implemente estos requisitos.

³Realizado en el laboratorio 6.2

5. Directrices con la gestión de un contenedor remoto

En el laboratorio 6.3 se realizaron actividades para configurar un servicio Web en un contenedor para acceder remotamente y desplegar este servicio en un computador remoto. En esta práctica, se propone desplegar un servicio de base de datos de manera remota. La motivación ha sido simplificar el trabajo, mientras se ilustre el proceso estándar en sistemas industriales. A continuación se describen con detalles la secuencia de ejecución.

1. Utilizamos dos computadores: **1A**, que es el local y gestiona las operaciones; y **1B**, que es el remoto. Esta misma notación se ha empleado en el laboratorio 6.3.
2. En el computador remoto (1B), hay que realizar las siguientes operaciones:

- Permitir el acceso remoto a las operaciones de LXD. Es necesario sustituir IP-B por la dirección IP del equipo 1B:

```
1B$ lxc config set core.https.address IP-B:8443
```

- Información para la acreditación en remoto. Es necesario definir una contraseña de acceso, que en este comando definimos como mypass:

```
1B$ lxc config set core.trust_password mypass
```

3. En el computador local (1A), hay que realizar las siguientes operaciones:

- Permitir el acceso remoto a las operaciones de LXD. Es necesario sustituir IP-A por la dirección IP del equipo 1A:

```
1A$ lxc config set core.https.address IP-A:8443
```

- Acreditarse en el sistema remoto. Esto permite al equipo 1A conectarse de manera remota al servicio LXD que se ejecuta en el equipo 1B. *remoto* es el nombre que le damos al remoto de LXD.

```
1A$ lxc remote add remoto IP-B:8443 \
--password mypass --accept-certificate
```

4. Configurar un *bridge* remoto. Este bridge ya existe en el equipo 1B (se crea por defecto al ejecutar el comando `lxd init --auto`), pero es necesario configurar su rango de direcciones IP.

```
1A$ lxc network set remoto:lxdb0 ipv4.address 134.3.0.1/24
```

```
1A$ lxc network set remoto:lxdb0 ipv4.nat true
```

5. Crear el contenedor de la base de datos db. Para esto existen dos alternativas:

- a) **Primera alternativa.** Crear el contenedor en el equipo local (1A) y posteriormente copiar este contenedor al equipo remoto (1B). Para ello, es necesario:

- Crear el contenedor en el equipo local
- Configurar el contenedor (dirección IP adecuada)
- Instalar los elementos necesarios: MongoDB y modificar el fichero `mongodb.conf`

Para copiar el contenedor ya configurado al equipo remoto podemos usar el siguiente comando. Esto copiará el contenedor de la base de datos al remoto, y mantendrá la configuración realizada.

```
1A$ lxc copy db remoto:db
```

- b) **Segunda alternativa.** Crear *remotamente* el contenedor directamente en el equipo remoto (1B). Para ello, es necesario (siempre desde el equipo local 1A):

- Crear el contenedor en el equipo remoto:

```
1A$ lxc init ubuntu2004 remoto:db
```

- Configurar la dirección IP del contenedor:

```
1A$ lxc config device override remoto:db eth0 \
ipv4.address=134.3.0.20
```

- Instalar el servicio de MongoDB en el contenedor de base de datos remoto.

```
1A$ lxc exec remoto:db -- apt update
```

```
1A$ lxc exec remoto:db -- apt install -y mongodb
```

6. Crear un proxy, para acceso remoto a las base de datos de manera remota:

```
1A$ lxc config device add remoto:db miproxy proxy \
listen=tcp:IP-B:27017 connect=tcp:134.3.0.20:27017
```

De esta manera, las peticiones enviadas a IP-B:27017 se redirigen al contenedor, a 134.3.0.20:27017.

5.1. Consideraciones para el despliegue remoto

Es necesario tener en cuenta los siguientes aspectos para que el despliegue del contenedor de base de datos db en remoto funcione correctamente.

- El contenedor de la base de datos remoto debe estar conectado al bridge lxdbr0 del equipo remoto (1B) y tener asignada la dirección IP 134.3.0.20.

- Cambiar la dirección IP de *MongoDB* en el fichero **rest_server.js** de la aplicación. Para ello, hay que buscar el texto:

```
await mongoose.connect('mongodb://134.3.0.20/bio_bbdd', .....
```

por el texto siguiente. Con ello, se indica a la aplicación la dirección IP donde se ejecuta *MongoDB*

```
await mongoose.connect('mongodb://IP-B/bio_bbdd', .....
```

- Cambiar la dirección IP de *MongoDB* en el fichero **md-seed-config.js** de la aplicación. Para ello, hay que buscar el texto:

```
const mongoURL = process.env.MONGO_URL || 'mongodb://10.0.0.20:27017/bio_bbdd';
```

y sustituirlo por el texto siguiente, con lo que se indica a la aplicación la dirección IP donde se debe probar la base de datos *MongoDB*.

```
const mongoURL = process.env.MONGO_URL || 'mongodb://IP-B:27017/bio_bbdd';
```

El fichero md-seed-config.js se emplea únicamente al realizar la instalación de NodeJS en los servidores, su uso se contiene dentro del fichero install.sh que se proporciona como parte de los laboratorios.

6. Partes opcionales adicionales

Como funcionalidades adicionales se propone incluir:

- Crear dinámicamente servidores con la aplicación. El objetivo es añadir servidores adicionales, suponiendo que hubiera aumentando la capacidad requerida. En concreto, se puede añadir la orden **enlarge** que cree e integre un servidor de la aplicación en un sistema.

- Emplear la aplicación que se haya desarrollado en la asignatura *Bases de Datos*. Se podrá modificar la página de la aplicación con los nombres del grupo. Se podrá demostrar que la aplicación usada es la desarrollada de uno de los miembros del grupo.

OjO: La aplicación se usa en BBDD (práctica 4). Si se integra, usa muchas bibliotecas que no son necesarias. Al generar el fichero de entrega es mayor de 20 MB

7. Recursos proporcionales e información adicional

Las prácticas se pueden desarrollar en distintos computadores:

- 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*. Esta aplicación se usó en el *Laboratorio 6.2*.
- La versión aconsejada de *Ubuntu* es la 20.04, que es la que se usa en los laboratorios.
- Aconseja usar los computadores en el laboratorio B-123 del Departamento, especialmente, para el desarrollo del despliegue y uso de un contenedor remoto.
- Si se desarrolla en un computador propio en Linux, para instalar *lxd*, ejecute las siguientes órdenes:

```
5 $ sudo apt update
$ sudo apt upgrade
$ sudo apt install -y lxd
$ sudo usermod -a -G lxd $USER
$ newgrp lxd
$ sudo lxd init --auto
```

8. Normas

Todas las entregas y prácticas que se realicen deben ser fruto del trabajo personal del alumno o grupo, aunque se fomentará la discusión y el trabajo en grupo para ayudar a entender mejor los problemas que se intentan resolver. La copia de entregas supondrá el suspenso de la asignatura de forma automática, tanto para quien copia como para quien se deja copiar.

En concreto, se puede:

- discutir los ejercicios con otros compañeros
- ayudar a otros a depurar sus ejercicios
- usar código publicado en el sitio web de la asignatura
- usar código publicado en otros sitios citando la procedencia, siempre que no sea la propia práctica

No se puede:

- copiar las prácticas de otro grupo, ni permitir la copia de las propias
- usar código publicado sin citar el origen