



## Práctica 2

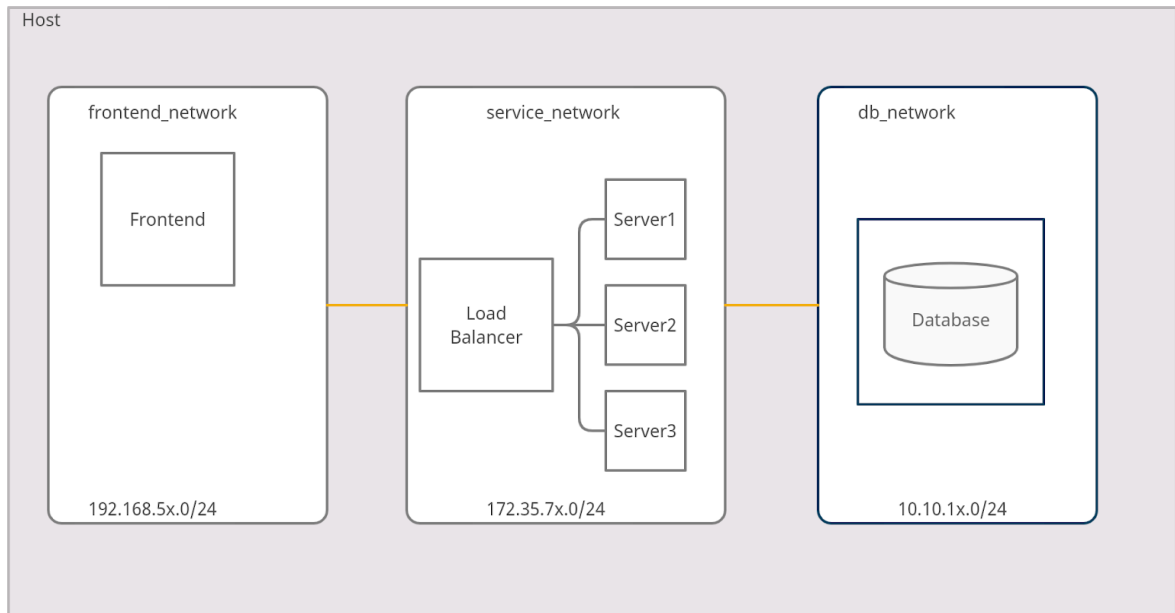
### Objetivos

- Conocer las herramientas provistas por los sistemas Linux para la creación y manejo de Virtual Networks.
- Conocer el dispositivo Bridge (Switch virtual) en Linux.
- Conocer las interfaces virtuales Veth.
- Comprender el funcionamiento de las redes virtuales y como estas permiten la comunicación entre host virtuales (Máquinas virtuales / Contenedores).
- Conocer los comandos provistos por Docker para la creación de redes virtuales usando el driver Bridge.
- Realizar agrupamiento de contenedores y segmentación de redes en Docker.
- Utilizar balanceo de carga por software para distribuir el tráfico entre contenedores.

### Descripción del problema

La Escuela de Ciencias y Sistemas de la Universidad de San Carlos de Guatemala desea reemplazar el sistema actual de envío de reportes de actividades para los alumnos que realizan sus prácticas finales. Dicho sistema se alojará en un único servidor en la nube. Se ha identificado que la mayoría de los practicantes espera a las fechas límites para realizar y enviar sus reportes, por lo que en dichas fechas se experimenta una caída en el sistema actual. Es por esto, que es de vital importancia que el nuevo sistema a desarrollar sea fácilmente escalable. Se plantea el uso de Docker para contenerizar las aplicaciones o servicios que conformen el nuevo sistema. Se le solicita a usted como estudiante de la escuela, realizar un demo funcional.

# Arquitectura



**Nota:** Para las direcciones de red a utilizar en cada Network, se utilizará el mismo mecanismo que se utilizó en la práctica anterior, es decir, se debe reemplazar la "x" en la dirección de red por el número de grupo. Si su grupo es de 2 dígitos, se utilizará el segundo dígito.

## db\_network

En esta network se alojarán los contenedores con el dbms. Los contenedores pertenecientes a dicha network solo pueden ser accedidos por contenedores de la service\_network. No se permite el acceso directo a través de contenedores de otras networks (excepto service\_network) ni tampoco es accesible desde el host. Inicialmente solo se contará con un contenedor de base de datos.

## service\_network

En esta network se alojan los contenedores que proveerán de servicios para interactuar con la base de datos a través de un api. Esta network también contendrá un balanceador de carga, el cual se encargará de distribuir el tráfico entre los distintos contenedores que contienen el api. Es importante resaltar que el acceso al api, sólo se puede realizar a través del balanceador de carga, es decir, ninguno de los servidores con el api puede ser accedidos de manera directa. Inicialmente se contará con 3 tres contenedores que provean el api.

## frontend\_network

En esta network se alojará el contenedor que proveerá del frontend, dicho frontend consumirá el api a través del balanceador de carga. Se hará uso de un solo contenedor con frontend.

## Base de datos

La base de datos a utilizar queda a discreción del estudiante. Se debe utilizar un volumen para tener persistencia de información.

## Servidor API

La tecnología para utilizar para realizar el api también queda a discreción del estudiante. Cada servidor debe tener una firma o identificador único, el cual adjuntará a cada respuesta para las peticiones que procese. Ejemplo: ***“Solicitud atendida por el servidor xxxxxxxx”***. Ya que son 3 servidores, se tomará como identificador/firma los carnés de los miembros del grupo.

***Se recomienda el uso de variables de entorno para manejar los identificadores de cada servidor sin necesidad de modificar el código para cada contenedor.***

## Frontend

La tecnología o herramientas para utilizar quedan a discreción del estudiante. Debido a que el objetivo de esta demo solo es testear la segmentación de redes en Docker, y el balanceo de carga, solo se solicita que el usuario pueda realizar las siguientes acciones:

- Enviar reporte: Un reporte estará conformado por los siguientes datos:
  - Carné del practicante que envía el reporte.
  - Nombre del estudiante.
  - Curso/proyecto al cual está asignado.
  - Cuerpo de reporte o mensaje.
- Listar reportes almacenados: Devuelve el listado de todos los reportes enviados por los practicantes, si recibe un carné como parámetro, devolverá sólo los reportes enviados por dicho practicante.
- Ver reporte individual: Permite ver un reporte de manera individual. Los datos para mostrar son:
  - Carné del practicante que envía el reporte.
  - Nombre del estudiante.
  - Curso/proyecto al cual está asignado.
  - Cuerpo de reporte o mensaje.
  - Servidor que proceso/ingreso dicho reporte.

## Pantalla de Ingreso de reporte

The screenshot shows a web application window titled "Ingreso reportes". Inside, there is a form titled "Ingreso de reportes de practicantes". The form contains four input fields: "Carnet", "Nombre", "Curso/proyecto", and "Cuerpo del reporte". The "Cuerpo del reporte" field is a larger text area with a small icon in the bottom right corner.

## Pantalla de Vista de reporte individual

The screenshot shows a web application window titled "Ver reporte". Inside, there is a form titled "Ingreso de reportes de practicantes". The form contains six input fields: "Carnet" (201515743), "Nombre" (Luis Perez), "Curso/proyecto" (Teoría de Sistemas 1), "Procesado por" (201564344(Servidor)), "Fecha" (12/03/2021), and "Cuerpo del reporte". The "Cuerpo del reporte" field is a larger text area with a small icon in the bottom right corner. Below the form, there is a message: "Solicitud atendida por el servidor "201614566"".

Nota: el valor del campo "Procesado por" es el identificador (carnet ) del servidor que guardó dicho reporte en la base de datos.

## Pantalla de listado de reportes

Si al presionar en el botón buscar el campo carné se encuentra vacío, se retornan todos los reportes almacenados en el sistema. Si el campo carné contiene un valor, se retorna solo los reportes que coincidan con dicho carné.



The screenshot shows a web application window titled "Listado reportes". Inside, there is a section titled "Listado de reportes" with a search form. The form has a label "Carnet" followed by a text input field and a "Buscar" button. Below the form is a table with 5 columns: "Carnet", "Nombre", "Proyecto", "Fecha", and "Servidor". The table contains one data row with the following values: "202012324", "Juan Prez", "Economía", "14/02/2021", and "201518647". At the bottom of the window, there is a status message: "Solicitud atendida por el servidor '201614566'" data-bbox="174 169 829 481"/>

Carnet	Nombre	Proyecto	Fecha	Servidor
202012324	Juan Prez	Economía	14/02/2021	201518647

*Solicitud atendida por el servidor "201614566"*

## Restricciones

- La práctica se realizará en los grupos ya formados.
- Todos los integrantes del grupo deben de tener conocimiento del desarrollo de la práctica.
- La práctica se debe realizar sobre un host con Linux, en un servidor en la nube.
- Es obligatorio el uso de Docker y Docker Compose.
- Para la calificación se debe de presentar la práctica en una computadora de los integrantes del grupo.
- Se debe de crear un repositorio de GitHub donde se irá actualizando el desarrollo de la práctica, el cual debe de contener como mínimo 2 commits por semana por parte de cada uno de los integrantes del grupo.
- Durante la calificación se preguntará información relevante de la práctica para comprobar la autoría de este.
- El manual técnico debe ser un pdf con el nombre **Practica2\_Manual\_#grupo.pdf**

## Penalizaciones

- Falta de seguimiento de desarrollo continuo por medio de Github tendrá una penalización del 10%.
- Falta de seguimiento de instrucciones conforme al método de entrega (nombre del repositorio) tendrá una penalización del 5%.
- Falta de puntualidad conforme a la entrega tendrá una penalización de la siguiente manera:
  - a. 1-10 minutos – 10%.
  - b. 11-59 minutos – 30%.
  - c. Pasados 60 minutos tendrá una nota de 0 y no se calificará.

## Observaciones

- La entrega se realizará por medio de **Github**, cada grupo deberá crear un repositorio con el nombre: **REDES2\_2S2021\_GRUPO#**, ejemplo: REDES2\_2S2021\_GRUPO3, y agregar a su auxiliar correspondiente como colaborador del mismo, para poder analizar su progreso y finalmente a partir del mismo repositorio realizar la calificación correspondiente. ***Este será el repositorio a utilizar a partir de ahora para la entrega de prácticas y proyectos, por lo que para cada práctica/proyecto, se deberá crear un directorio con el nombre de la práctica o proyecto, ejemplo:***
  - **Practica 1**
  - **Practica 2**
  - **.....**
  - **Proyecto 1**
- Dentro del repositorio anterior se debe crear un directorio llamado “Práctica 2”, el cual debe contener todos los archivos utilizados en la práctica (Código fuente, Dockerfiles, archivos yaml de compose, etc.)
- Además de tener a su auxiliar como colaborador del repositorio para tener un control y orden de las personas que entreguen deberán de colocar el Link de su repositorio en la Tarea que cada auxiliar asignará en su plataforma correspondiente.
- Fecha y hora de entrega: **Sábado, 25 de septiembre, antes de las 23:59 horas.**
- **Las copias serán penalizadas con una nota de 0 y castigadas según lo indique el reglamento.**