

ACTIVITAT AVALUABLE PF**Mòdul:** MP08- Desplegament d'aplicacions web**UF:** UF3 – Aplicacions web**Professor:** Albert Guardiola**Data límit d'entrega:** 27/4/25**Mètode d'entrega:** Per mitjà del Clickedu de l'assignatura. Les activitats entregades més enllà de la data límit només podran obtenir una nota de 5.**Instruccions:** S'ha d'entregar un únic document amb el nom:***MP08-UF3-PF-Nom_Alumne.doc (o pdf)***

Es valorarà la presentació.

Resultats de l'aprenentatge:

RA1. Verifica l'execució d'aplicacions web comprovant els paràmetres de configuració de serveis de xarxa.

Para realizar esta actividad, vamos a utilizar el siguiente repositorio, del cual deberéis hacer un *fork* en vuestra cuenta de GitHub personal:

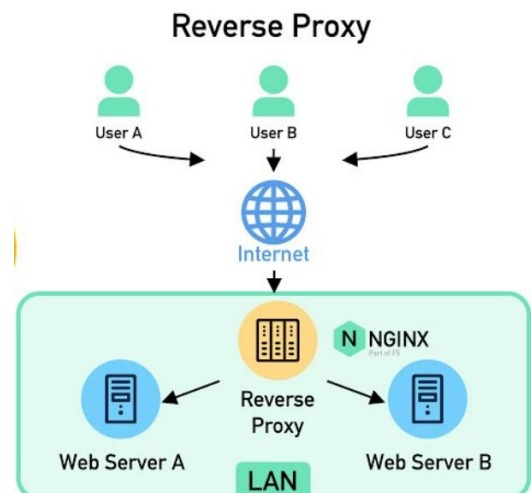
<https://github.com/hayk96/trafficlight-docker-challenge>

1ª parte – Despliegue de un tres aplicaciones web tras un proxy inverso (6 puntos)

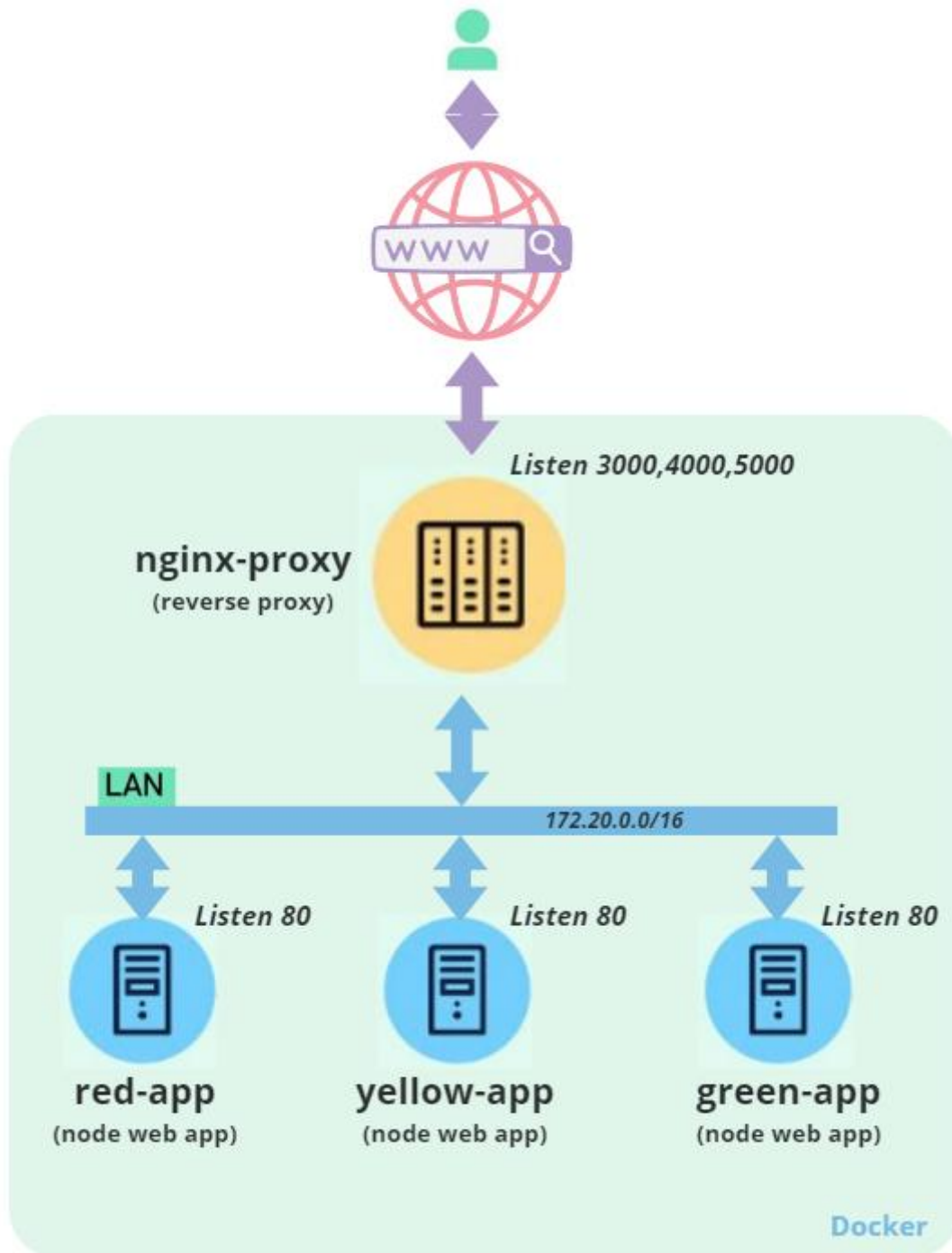
Tarea 0. En esta primera parte de la práctica, se trata de completar las tareas 1 y 2 del *Trafficlight docker challenge* de utilizar contenedores Docker para desplegar tres aplicaciones web (*red-app*, *yellow-app* y *green-app*) tras un proxy reverso, de manera que las tres esten accesibles tras un proxy reverso, en las URLs <http://localhost:3000>, <http://localhost:4000> y <http://localhost:5000>, respectivamente.

Antes de empezar con la compilación de las imágenes en Docker, familiarízate con el concepto de proxy inverso y sus casos de uso:

<https://www.youtube.com/watch?v=4NB0NDtOwIQ>



Tarea 1. Sigue las instrucciones detalladas en las tareas 1 y 2 del challenge para desplegar la siguiente infraestructura.

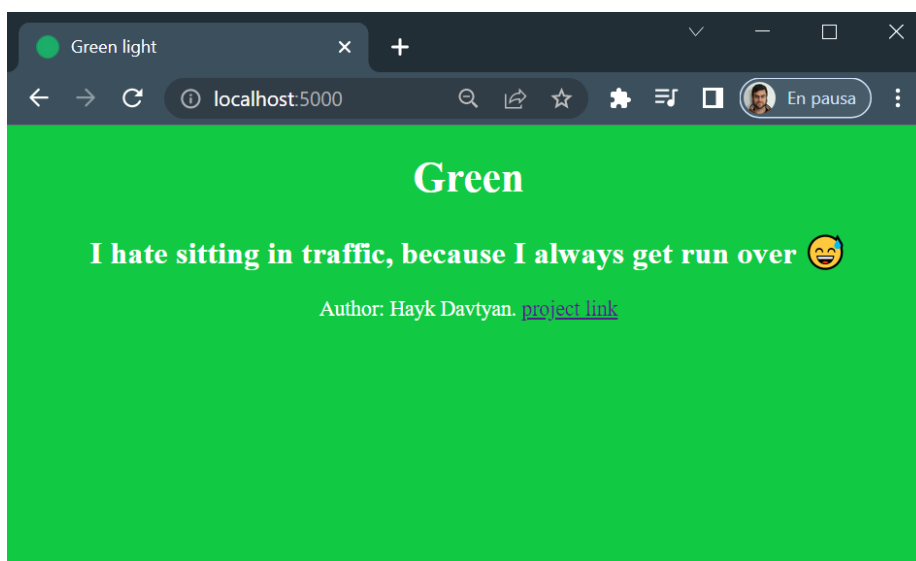
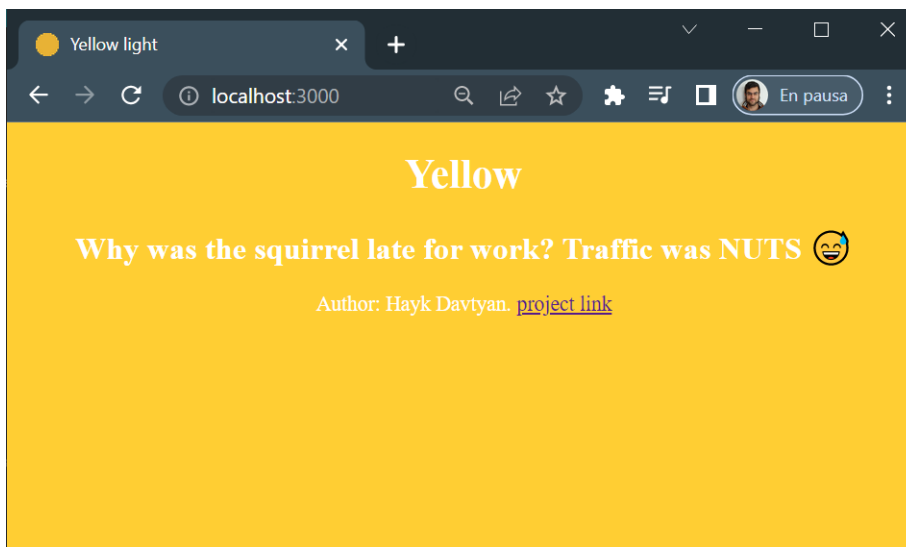
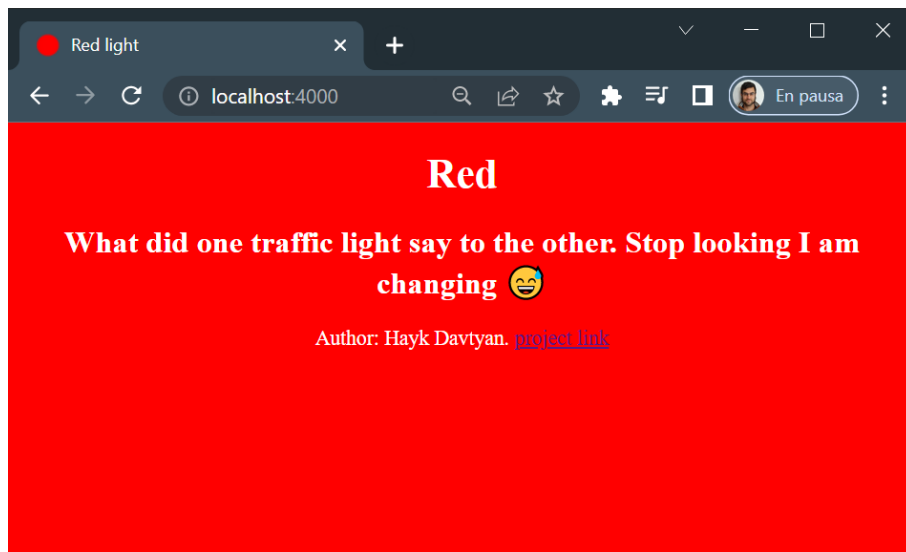


Como se ve, la aplicación consta de cuatro contenedores Docker:

- Tres de ellos incluyen una aplicación node que sirve una página web de chistes.
- El cuarto incluye un servidor web nginx configurado para funcionar como proxy inverso.

Los cuatro contenedores están conectados entre sí a través de una red interna *traffic-light*.

Estas son la respuestas esperadas a las peticiones al proxy inverso:



(Los chistes son aleatorios, malos y, en algún caso, tremendamente misóginos. Sentiros libres de borrarlos o modificarlos)

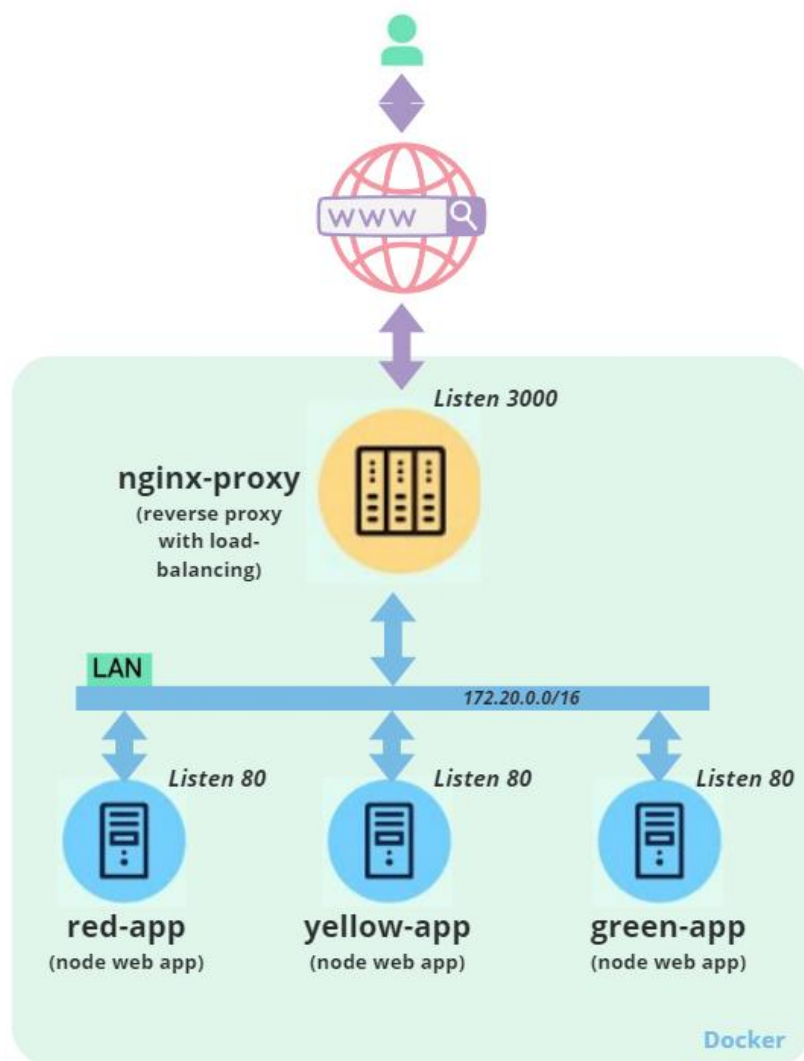
2ª parte – Despliegue de un tres aplicaciones web tras un proxy inverso como balanceador de carga (4 puntos)

Tarea 00. Consulta el siguiente vídeo para familiarizarte con el concepto de balanceo de carga:

<https://www.youtube.com/watch?v=a41jxGP9lc8>

Este es uno de los usos más comunes del servidor nginx y, concretamente, de su instalación como proxy inverso. Habitualmente, el balanceo de carga se da entre distintas instancias de la misma aplicación web, para repartir el tráfico entrante, pero sirviendo exactamente el mismo contenido. En este caso, y para isualizar mejor qué servidor está sirviendo la web, utilizaremos las aplicaciones *red-app*, *yellow-app* y *green-app*.

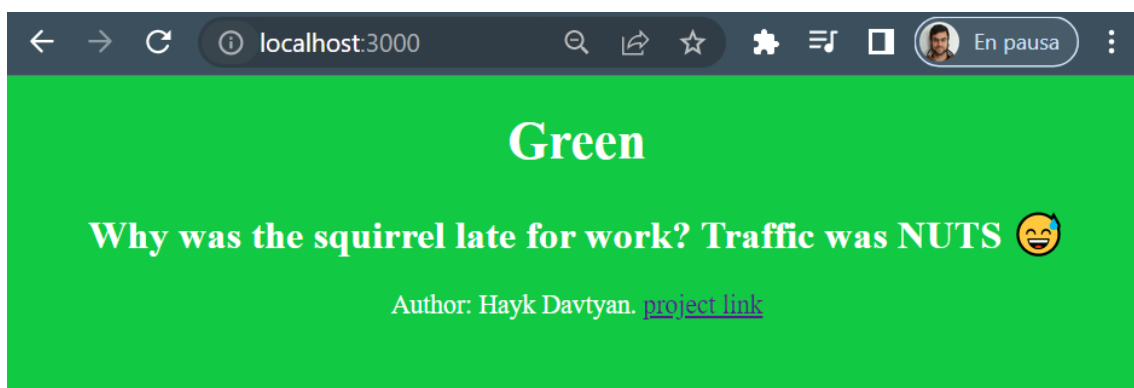
Tarea 2. Sigue las instrucciones detalladas en la tarea 3 del challenge para desplegar la siguiente insfraestructura.



Observa que han cambiado dos cosas:

- Ahora el proxy sólo escucha el puerto 3000.
- El proxy está configurado como balanceador de carga, de manera que decidirá, en función del modo de balanceo seleccionado, qué aplicación web servir en cada momento (por defecto, irás sirviendo una de las tres alternativamente).

Estas son la respuestas esperadas a las peticiones al proxy inverso (observa que la URL es siempre la misma):



La entrega de la PF ha de consistir en el enlace al repositorio en el que se ha solucionado el reto, es decir, en el que se incluyan **los Dockerfile y los ficheros de configuración de nginx** necesarios para desplegar el reto, así como **los docker-compose.yml** que levantan una y otra versión de la actividad (proxy inverso y balanceador de carga).

LA CORRECCIÓN SE HARÁ EJECUTANDO LOS DOCKER-COMPOSE.YML