

Servidores Web de Altas Prestaciones

PRÁCTICA 1

Servidores web y almacenamiento



**UNIVERSIDAD
DE GRANADA**

Grupo Pequeño	A2
----------------------	----

Garvín Pérez, Santiago

Grado en Ingeniería Informática

Curso 2023-2024

ÍNDICE

Tareas Básicas	3
B1. Configuración del Entorno	3
- Análisis propuesta IA	3
B2. Creación del Dockerfile	4
- Análisis propuesta IA	4
B3. Uso de Docker Compose	5
- Análisis propuesta IA	8
B4. Despliegue y verificación de Contenedores	8
- Análisis propuesta IA	10
B5. Pruebas Básicas	10
Tareas Avanzadas	11
A1. Personalización del Dockerfile	11
- Análisis propuesta IA	12
A2. Creación de contenedores con otros servidores web	13
- Análisis propuesta IA	13
A3. Gestión Avanzada de Redes	14
- Análisis propuesta IA	15
A4. Automatización con Scripts	16
- Análisis propuesta IA	20

Nota: Para la mayor parte de las consultas realizadas he utilizado la inteligencia artificial de Google, Gemini 1.5 Pro (<https://aistudio.google.com>) junto con ChatGPT 3.5 (<https://chat.openai.com>) de OpenAI. Con la ayuda de ambos he ido contrastando la información que iba obteniendo.

Tareas Básicas

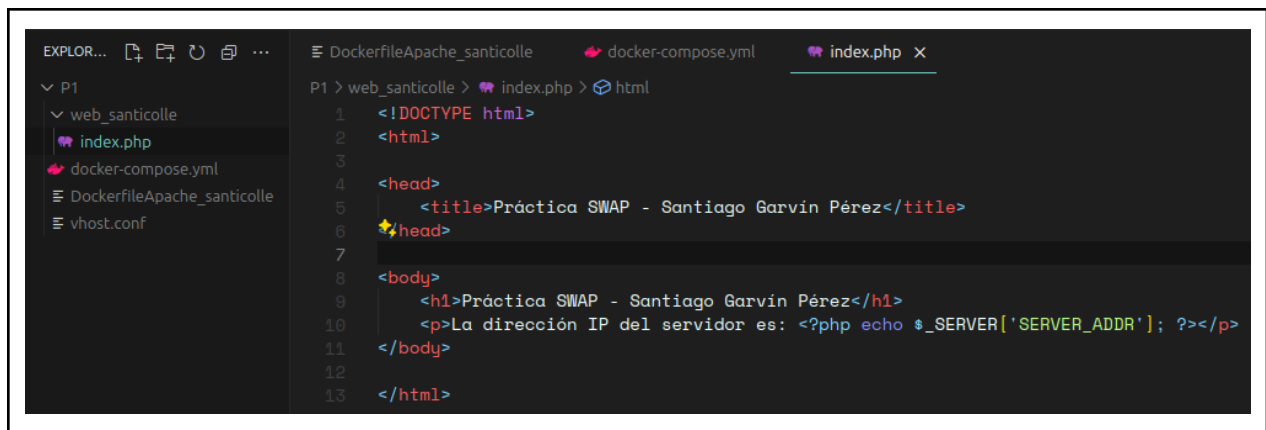
B1. Configuración del Entorno

- Instalar Docker y Docker Compose si aún no están instalados.

Para la instalación de Docker no he tenido problema, con ayuda de Gemini he añadido los repositorios de Docker junto con su llave GPG y certificado, después he ejecutado

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

- Crear el directorio **web_usuarioUGR** y el archivo **index.php**.

A screenshot of a code editor interface. The left sidebar shows a file explorer with a project named 'P1' containing a directory 'web_santicolle' with files 'index.php', 'docker-compose.yml', 'DockerfileApache_santicolle', and 'vhost.conf'. The main editor area shows the content of 'index.php' with the following HTML code:

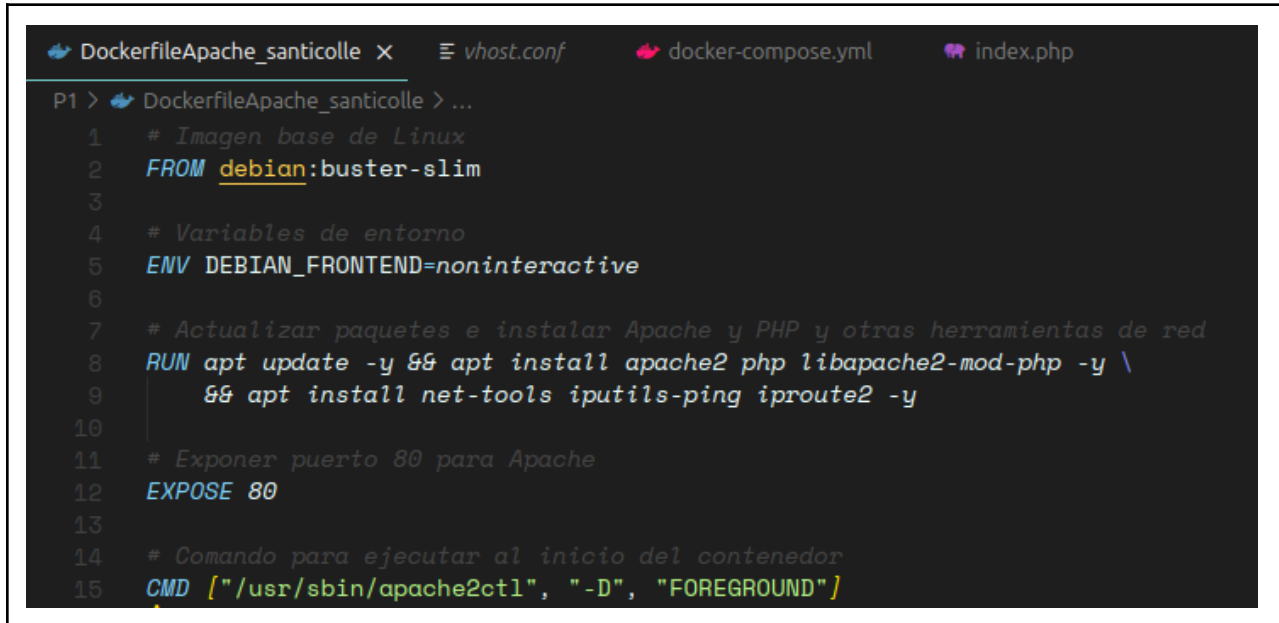
```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Práctica SWAP - Santiago Garvín Pérez</title>
6 </head>
7
8 <body>
9   <h1>Práctica SWAP - Santiago Garvín Pérez</h1>
10  <p>La dirección IP del servidor es: <?php echo $_SERVER['SERVER_ADDR']; ?></p>
11 </body>
12
13 </html>
```

- Análisis propuesta IA

Para escribir el código de index.php sí que me ha sido útil saber la sentencia de php que nos da la dirección IP. Esto sería lo único que he hecho con IA en este apartado.

B2. Creación del Dockerfile

- Escribir un Dockerfile **DockerfileApache_usuarioUGR** que use una imagen base de Linux, e instale Apache, PHP y herramientas de red.



```
P1 > DockerfileApache_santicolle > ...
1  # Imagen base de Linux
2  FROM debian:buster-slim
3
4  # Variables de entorno
5  ENV DEBIAN_FRONTEND=noninteractive
6
7  # Actualizar paquetes e instalar Apache y PHP y otras herramientas de red
8  RUN apt update -y && apt install apache2 php libapache2-mod-php -y \
9      && apt install net-tools iputils-ping iproute2 -y
10
11 # Exponer puerto 80 para Apache
12 EXPOSE 80
13
14 # Comando para ejecutar al inicio del contenedor
15 CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Nota: He modificado la imagen base a ubuntu:jammy

- Análisis propuesta IA

En este apartado la IA me ha ayudado con los comandos para exponer los puertos y sobre las herramientas que se suelen usar para hacer análisis de la red. Estas dos están también en las diapositivas de la primera práctica pero de esta forma ha sido más rápida la obtención del resultado, aún así he contrastado los dos resultados para ver cual era el más acertado. Para ejecutar el comando final he tenido que revisarlo porque no me lo proporcionaba correctamente, diría que era por no utilizar la ruta absoluta..

B3. Uso de Docker Compose

- Escribir un archivo **docker-compose.yml** que defina la construcción de la imagen **usuarioUGR-apache-image:p1** y la creación de los 8 contenedores **webX**.

docker-compose.yml

```
version: '3.1'

services:
  santicolle-apache:
    image: santicolle-apache-image:p1
    build:
      context: .
      dockerfile: DockerfileApache_santicolle
    volumes:
      - ./web_santicolle:/var/www/html/
    ports:
      - "8090:80"
    networks:
      red_web:
        ipv4_address: 192.168.10.10
      red_servicios:
        ipv4_address: 192.168.20.10
    hostname: servidor-web

  web1:
    image: santicolle-apache-image:p1
    volumes:
      - ./web_santicolle:/var/www/html/
    ports:
      - "8080:80"
    networks:
      red_web:
        ipv4_address: 192.168.10.2
      red_servicios:
        ipv4_address: 192.168.20.2
    hostname: servidor-web-1

  web2:
    image: santicolle-apache-image:p1
    volumes:
      - ./web_santicolle:/var/www/html/
    ports:
      - "8081:80"
    networks:
      red_web:
```

ipv4_address: 192.168.10.3
red_servicios:
ipv4_address: 192.168.20.3
hostname: servidor-web-2

web3:

image: *santicolle-apache-image:pl*
volumes:
- ./web_santicolle:/var/www/html/
ports:
- "8082:80"
networks:
red_web:
ipv4_address: 192.168.10.4
red_servicios:
ipv4_address: 192.168.20.4
hostname: servidor-web-3

web4:

image: *santicolle-apache-image:pl*
volumes:
- ./web_santicolle:/var/www/html/
ports:
- "8083:80"
networks:
red_web:
ipv4_address: 192.168.10.5
red_servicios:
ipv4_address: 192.168.20.5
hostname: servidor-web-4

web5:

image: *santicolle-apache-image:pl*
volumes:
- ./web_santicolle:/var/www/html/
ports:
- "8084:80"
networks:
red_web:
ipv4_address: 192.168.10.6
red_servicios:
ipv4_address: 192.168.20.6
hostname: servidor-web-5

web6:

image: *santicolle-apache-image:pl*
volumes:

- ./web_santicolle:/var/www/html/

ports:

- "8085:80"

networks:

red_web:

ipv4_address: 192.168.10.7

red_servicios:

ipv4_address: 192.168.20.7

hostname: servidor-web-6

web7:

image: *santicolle-apache-image:pl*

volumes:

- ./web_santicolle:/var/www/html/

ports:

- "8086:80"

networks:

red_web:

ipv4_address: 192.168.10.8

red_servicios:

ipv4_address: 192.168.20.8

hostname: servidor-web-7

web8:

image: *santicolle-apache-image:pl*

volumes:

- ./web_santicolle:/var/www/html/

ports:

- "8087:80"

networks:

red_web:

ipv4_address: 192.168.10.9

red_servicios:

ipv4_address: 192.168.20.9

hostname: servidor-web-8

networks:

red_web:

ipam:

config:

- subnet: 192.168.10.0/24

red_servicios:

ipam:

config:

- subnet: 192.168.20.0/24

- Análisis propuesta IA

Para hacer esta parte le he preguntado a Gemini como poner un Dockerfile que tenga un nombre distinto al predeterminado y para saber cómo escribir los parámetros de build, me ha dado algunas respuestas erróneas como poner el build a nivel de services, fuera de cada servicio, incluso ponerlo dentro de cada una de las web, en ese momento he contrastado con Chat GPT y me ha recomendado poner todo el build dentro de santicolle-apache que es el contenedor principal, y que el resto de imágenes lo usen como base.

B4. Despliegue y verificación de Contenedores

- Ejecutar **docker-compose up** para iniciar los contenedores.

```
~/Escritorio/SWAP/P1 docker compose up --build --force-recreate ✓ < base Py
[+] Building 2.1s (7/7) FINISHED docker:desktop-linux
=> [santicolle-apache internal] load build definition from DockerfileApache_santicolle 0.0s
=> => transferring dockerfile: 503B 0.0s
=> [santicolle-apache internal] load metadata for docker.io/library/debian:buster-slim 1.9s
=> [santicolle-apache auth] library/debian:pull token for registry-1.docker.io 0.0s
=> [santicolle-apache internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [santicolle-apache 1/2] FROM docker.io/library/debian:buster-slim@sha256:4620251e86cea652ef2746c8a 0.0s
=> CACHED [santicolle-apache 2/2] RUN apt update -y && apt install apache2 php libapache2-mod-php -y 0.0s
=> [santicolle-apache] exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:dd51a48383a1592406655fb12cb1d81f54cd4f7df4d4a06c3fafb657debf2e28 0.0s
=> => naming to docker.io/library/santicolle-apache-image:p1 0.0s
[+] Running 11/11
✓ Network p1_red_web Created 0.2s
✓ Network p1_red_servicios Created 0.1s
✓ Container p1-web3-1 Created 0.4s
✓ Container p1-web2-1 Created 0.5s
✓ Container p1-santicolle-apache-1 Created 0.5s
✓ Container p1-web4-1 Created 0.5s
✓ Container p1-web1-1 Created 0.5s
✓ Container p1-web5-1 Created 0.5s
✓ Container p1-web7-1 Created 0.5s
✓ Container p1-web6-1 Created 0.5s
✓ Container p1-web8-1 Created 0.5s
Attaching to santicolle-apache-1, web1-1, web2-1, web3-1, web4-1, web5-1, web6-1, web7-1, web8-1
web7-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.10.8. Set the 'ServerName' directive globally to suppress this message
web4-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.5. Set the 'ServerName' directive globally to suppress this message
web6-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.7. Set the 'ServerName' directive globally to suppress this message
web1-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.2. Set the 'ServerName' directive globally to suppress this message
web5-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.6. Set the 'ServerName' directive globally to suppress this message
web2-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.10.3. Set the 'ServerName' directive globally to suppress this message
web8-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.9. Set the 'ServerName' directive globally to suppress this message
web3-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.20.4. Set the 'ServerName' directive globally to suppress this message
santicolle-apache-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 192.168.10.10. Set the 'ServerName' directive globally to suppress this message
```


- Usar **docker ps** para asegurarse de que todos los contenedores están en ejecución.

```
~/Escritorio/SWAP/P1 docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b045c3f105ac	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8090->80/tcp	p1-santicolle-apache-1
102a26fb5bab	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8087->80/tcp	p1-web8-1
8b9f62774d4d	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8080->80/tcp	p1-web1-1
30eb838f1968	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8083->80/tcp	p1-web4-1
17da981cc815	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8085->80/tcp	p1-web6-1
ee7acf32c2cf	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8084->80/tcp	p1-web5-1
c9ea92c2645f	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8086->80/tcp	p1-web7-1
9003b2852a3b	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8081->80/tcp	p1-web2-1
b671ca791cf5	santicolle-apache-image:pl	"/usr/sbin/apache2ct..."	23 minutes ago	Up 3 minutes	0.0.0.0:8082->80/tcp	p1-web3-1

- Verificar que cada contenedor tiene una dirección IP asignada en las redes **red_web** y **red_servicios**.

```
docker network inspect p1_red_web --format='{{range .Containers}}{{.Name}} - {{.IPv4Address}}\n'{{end}}'
```

```
docker network inspect p1_red_servicios --format='{{range .Containers}}{{.Name}} - {{.IPv4Address}}\n'{{end}}'
```

```
~/E/SWAP/P1 docker network inspect p1_red_web --format='{{range .Containers}}{{.Name}} - {{.IPv4Address}}\n'{{end}}'
```

p1-web8-1 - 192.168.10.9/24
p1-web6-1 - 192.168.10.7/24
p1-web2-1 - 192.168.10.3/24
p1-santicolle-apache-1 - 192.168.10.10/24
p1-web3-1 - 192.168.10.4/24
p1-web7-1 - 192.168.10.8/24
p1-web5-1 - 192.168.10.6/24

```
~/Escritorio/SWAP/P1 docker network inspect p1_red_servicios --format='{{range .Containers}}{{.Name}} - {{.IPv4Address}}\n'{{end}}'
```

p1-web8-1 - 192.168.20.9/24
p1-web6-1 - 192.168.20.7/24
p1-web4-1 - 192.168.20.5/24
p1-web1-1 - 192.168.20.2/24
p1-web2-1 - 192.168.20.3/24
p1-santicolle-apache-1 - 192.168.20.10/24
p1-web3-1 - 192.168.20.4/24
p1-web5-1 - 192.168.20.6/24

```
~/Escritorio/SWAP/P1 docker compose exec web1 /bin/bash
root@servidor-web-1:/# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=64 time=3.72 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=64 time=0.659 ms
64 bytes from 192.168.10.3: icmp_seq=3 ttl=64 time=0.110 ms
64 bytes from 192.168.10.3: icmp_seq=4 ttl=64 time=0.119 ms
^C
--- 192.168.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 69ms
rtt min/avg/max/mdev = 0.110/1.152/3.722/1.500 ms
root@servidor-web-1:/#
```

- Análisis propuesta IA

Para lo único que he utilizado la IA en este caso es para darle formato a la salida que me proporciona *inspect* y así he podido verlo con una estructura más reducida. Por mi propia cuenta no hubiera sabido hacerlo pero puedo llegar a entender lo que hacen esas sentencias. Esto me lo ha dado Chat GPT, sin embargo, Gemini me había dado como respuesta hacer un *inspect* de cada uno de los contenedores, dándome mucha más información irrelevante.

B5. Pruebas Básicas

- Acceder a la página web de cada contenedor usando su dirección IP y verificar que muestra la información correcta.



Tareas Avanzadas

A1. Personalización del Dockerfile

- Modificar el Dockerfile para incluir configuraciones personalizadas de Apache o PHP.

Este es mi archivo de configuración **php.ini** que he creado para pasarle a php.

php.ini

```
; php.ini

; Configuración de memoria
memory_limit = 256M

; Configuración de errores
error_reporting = E_ALL & ~E_NOTICE
display_errors = On

; Configuración de archivos
file_uploads = On
upload_max_filesize = 10M

; Configuración de sesiones
session.save_path = "/tmp"

; Zona horaria
date.timezone = "Europe/Madrid"
```

He creado un pequeño script para mostrar la información de configuración de php para ver si se habían aplicado los cambios y lo he puesto en la carpeta de web_santicolle ya que está enlazada con bind mount.

```
<?php
phpinfo();
```

Esta es la línea que he añadido al **Dockerfile**:

```
COPY config_php/php.ini /etc/php/7.3/apache2/php.ini
```

La configuración se ha aplicado correctamente, estas son las pruebas de ello:

max_input_vars	1000	1000
max_multipart_body_parts	-1	-1
memory_limit	256M	256M
Default timezone	Europe/Madrid	

- Análisis propuesta IA

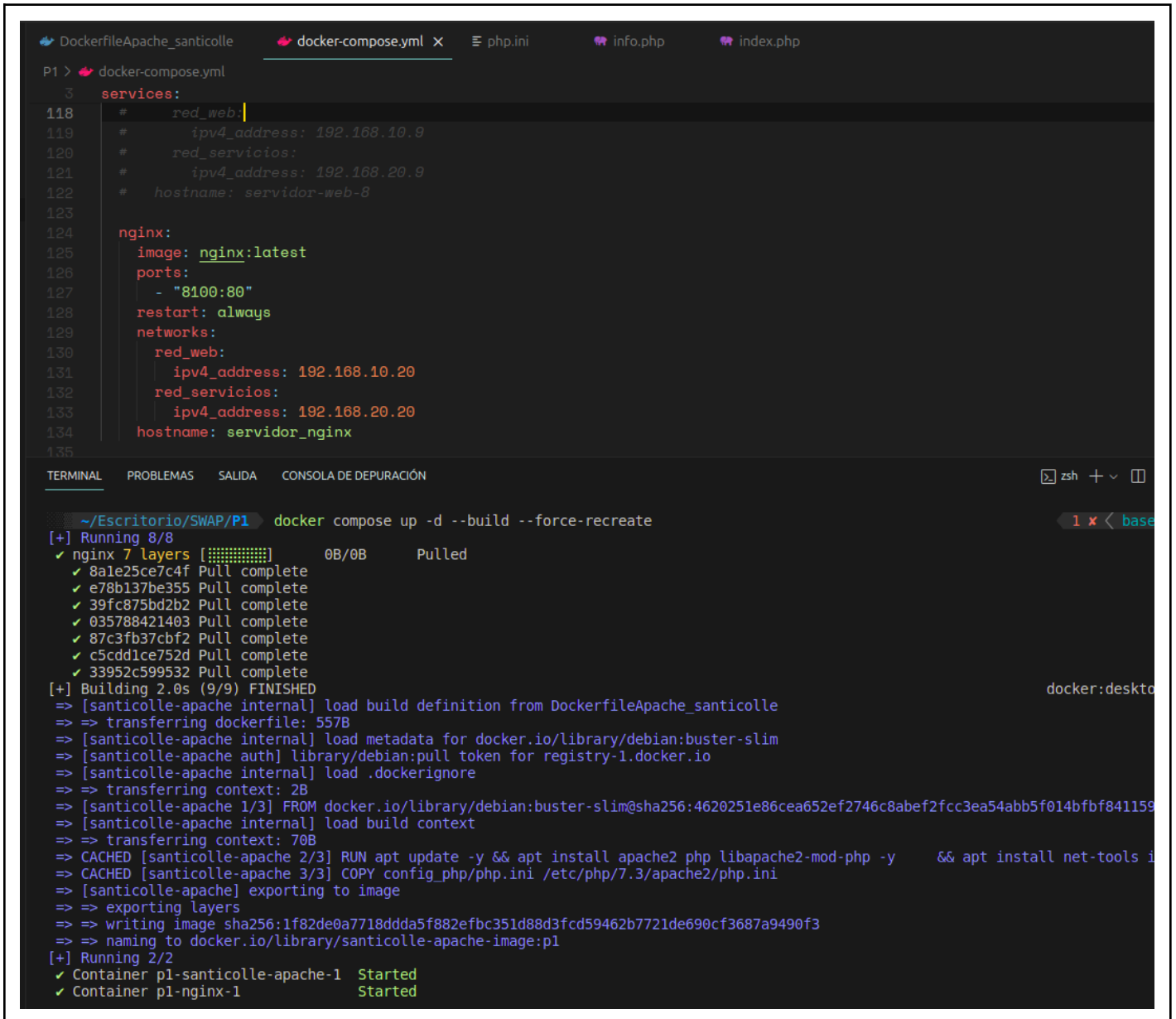
Gemini me ha dado unas cuantas respuestas erróneas como la siguiente:

```
=> ERROR [santicolle-apache 3/3] RUN cat config_php/php.ini >>
/etc/php/7.3/apache2/php.ini          0.3s
-----
> [santicolle-apache 3/3] RUN cat config_php/php.ini >>
/etc/php/7.3/apache2/php.ini:
0.227 cat: config_php/php.ini: No such file or directory
-----
```

Por lo que he hecho la misma pregunta con el error a Chat GPT y este si me ha sabido responder bien. Para ello he copiado directamente el archivo con la orden que he mencionado anteriormente en el Dockerfile. Mi intención era añadir lo que tuviera el archivo de mi local php.ini al archivo de configuración de php del contenedor pero creo que es mejor hacer un copy de todo el archivo para que no haya inconsistencias ni elementos duplicados dentro de la configuración.

A2. Creación de contenedores con otros servidores web

- Crear contenedores con otros servidores web (nginx, lighttpd, etc.)



```
DockerfileApache_santicolle  docker-compose.yml x  php.ini  info.php  index.php
P1 > docker-compose.yml
3  services:
118  #   red_web:
119  #       ipv4_address: 192.168.10.9
120  #   red_servicios:
121  #       ipv4_address: 192.168.20.9
122  #   hostname: servidor-web-8
123
124  nginx:
125  image: nginx:latest
126  ports:
127  - "8100:80"
128  restart: always
129  networks:
130  red_web:
131  ipv4_address: 192.168.10.20
132  red_servicios:
133  ipv4_address: 192.168.20.20
134  hostname: servidor_nginx
135

TERMINAL  PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN
~/Escritorio/SWAP/P1  docker compose up -d --build --force-recreate
[+] Running 8/8
  ✓ nginx 7 layers [■■■■■■■■] 0B/0B  Pulled
  ✓ 8a1e25ce7c4f Pull complete
  ✓ e78b137be355 Pull complete
  ✓ 39fc875bd2b2 Pull complete
  ✓ 035788421403 Pull complete
  ✓ 87c3fb37cbf2 Pull complete
  ✓ c5cdd1ce752d Pull complete
  ✓ 33952c599532 Pull complete
[+] Building 2.0s (9/9) FINISHED
=> [santicolle-apache internal] load build definition from DockerfileApache_santicolle
=> => transferring dockerfile: 557B
=> [santicolle-apache internal] load metadata for docker.io/library/debian:buster-slim
=> [santicolle-apache auth] library/debian:pull token for registry-1.docker.io
=> [santicolle-apache internal] load .dockerignore
=> => transferring context: 2B
=> [santicolle-apache 1/3] FROM docker.io/library/debian:buster-slim@sha256:4620251e86cea652ef2746c8abef2fcc3ea54abb5f014bfbf841159
=> [santicolle-apache internal] load build context
=> => transferring context: 70B
=> CACHED [santicolle-apache 2/3] RUN apt update -y && apt install apache2 php libapache2-mod-php -y && apt install net-tools i
=> CACHED [santicolle-apache 3/3] COPY config_php/php.ini /etc/php/7.3/apache2/php.ini
=> [santicolle-apache] exporting to image
=> => exporting layers
=> => writing image sha256:1f82de0a7718dda5f882efbc351d88d3fcd59462b7721de690cf3687a9490f3
=> => naming to docker.io/library/santicolle-apache-image:p1
[+] Running 2/2
  ✓ Container p1-santicolle-apache-1 Started
  ✓ Container p1-nginx-1 Started
```

He comentado los servidores web para estas pruebas de manera que el proceso de lanzamiento de contenedores fuera un poco más rápido.

- Análisis propuesta IA

En este apartado no he tenido que utilizar la IA, aún así le he pedido alguna recomendación de mejora del archivo docker-compose.yml pero no me ha proporcionado información que fuera de utilidad.

A3. Gestión Avanzada de Redes

- Configurar reglas específicas de enrutamiento o restricciones de acceso entre las dos redes red_web y red_servicios.

docker-compose.yml

```
services:
  santicolle-apache:
    image: santicolle-apache-image:pl
    build:
      context: .
      dockerfile: DockerfileApache_santicolle
    volumes:
      - ./web_santicolle:/var/www/html/
    ports:
      - "8080:80"
    cap_add:
    - NET_ADMIN
    user: root
    networks:
      red_web:
        ipv4_address: 192.168.10.10
      red_servicios:
        ipv4_address: 192.168.20.10
    hostname: servidor-web

  nginx:
    image: nginx:latest
    ports:
      - "8100:80"
    restart: always
    networks:
      red_web:
        ipv4_address: 192.168.10.20
# He quitado el acceso a la red de servicios
    hostname: servidor_nginx
```

Dockerfile

```
CMD ["/bin/sh", "-c", "service apache2 restart; \  
  /usr/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT; \  
  /usr/sbin/iptables -A INPUT -p tcp --dport 443 -j ACCEPT; \  
  /usr/sbin/iptables -A INPUT -p tcp --dport 8080 -j ACCEPT; \  
  /usr/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT; \  
  /usr/sbin/iptables -A INPUT -p tcp -j DROP; \  
  /usr/sbin/apache2ctl -D FOREGROUND \  
  "]
```

Nota: Al poner los comandos de iptables en RUN me daba error ya que no reconoció el argumento -dport, por lo que lo he puesto en CMD y no me ha dado problemas para reconocerlo.

He creado un sistema de cortafuegos con iptables, el problema es que no se puede hacer una configuración personalizada para cada contenedor, sino que todos tendrán abiertos los mismos puertos abiertos.

- Análisis propuesta IA

He tenido bastantes dudas con respecto a cómo poner las restricciones de que solo se pueda conectar a la red web a través de ciertos puertos. La IA de primeras me había dado como respuesta que pusiera la restricción en la web de servicios y, según lo que tengo entendido, esa red se usará como red de interconexión entre las bases de datos y los servidores web. Esto es lo que tenía pensado poner pero esta configuración NO funciona:

docker-compose.yml

```
networks:  
  red_web:  
    ipam:  
      config:  
        - subnet: 192.168.10.0/24  
  policies:  
    - source: red_web  
      protocol: tcp  
      ports:  
        - "443"  
        - "80"  
        - "22"  
    - source: red_web  
      protocol: udp  
      ports:  
        - "53"
```

```
red_servicios:
  ipam:
    config:
      - subnet: 192.168.20.0/24
  policies:
    - policy: egress
      to: red_web
      ports: []
```

La solución que he puesto en el Dockerfile también ha sido generada con IA y no creo que sea la más acertada pero la otra opción posible era configurar un proxy inverso con Nginx, lo cual ya es un poco más complicado.

A4. Automatización con Scripts

- Crear scripts para tareas de mantenimiento automatizado, como limpieza de logs, monitoreo de la salud del contenedor, o actualizaciones automáticas de paquetes.

He creado una carpeta con un archivo maintenance.sh donde he puesto las líneas de código que quería programar con crontab.

maintenance.sh

```
#!/bin/bash

# Actualizar paquetes
apt-get update
apt-get upgrade -y
apt-get clean

# Limpiar logs
find /var/log/cron -type f -ctime +30 -exec rm {} \;
```

Para ejecutar este script diariamente he modificado la línea de CMD del archivo Dockerfile.

Dockerfile

```
# Comprobación automática de salud
HEALTHCHECK --interval=5s --timeout=2s --retries=3 CMD curl -f
http://localhost:80/ || exit 1

CMD ["/bin/sh", "-c", "service apache2 restart; \
```



```
echo '@daily /etc/cron.d/maintenance.sh' | crontab -; \  
cron -f; \  
/usr/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT; \  
/usr/sbin/iptables -A INPUT -p tcp --dport 443 -j ACCEPT; \  
/usr/sbin/iptables -A INPUT -p tcp --dport 8080 -j ACCEPT; \  
/usr/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT; \  
/usr/sbin/iptables -A INPUT -p tcp -j DROP; \  
/usr/sbin/apache2ctl -D FOREGROUND;"]
```

En esta parte del Dockerfile he puesto también la instrucción para comprobar el estado del contenedor.

docker-compose.yml

```
services:  
  santicolle-apache:  
    image: santicolle-apache-image:pl  
    volumes:  
      - ./web_santicolle:/var/www/html/  
    build:  
      context: .  
      dockerfile: DockerfileApache_santicolle  
    ports:  
      - "8090:80"  
    cap_add:  
      - NET_ADMIN  
    user: root  
    restart: always
```

- Escribir scripts para automatizar la creación de contenedores o la configuración de la red.

redes.sh

```
#!/bin/bash

# Check if network name and IP address are provided
if [ -z "$1" ] || [ -z "$2" ]; then
    echo "Error: Network name and IP address are required."
    exit 1
fi

# Set network name and IP address from arguments
network_name=$1
ip_address=$2

# Create the network with the specified IP address and subnet mask
docker network create --subnet $ip_address $network_name
```

contenedores.sh

```
#!/bin/bash

# Check if container name and image name are provided
if [ -z "$1" ] || [ -z "$2" ]; then
    echo "Error: Container name and image name are required."
    exit 1
fi

# Set container name and image name from arguments
container_name=$1
image_name=$2

# Check if image exists locally
if [[ "$(docker images -q $image_name 2> /dev/null)" == "" ]]; then
    echo "Image '$image_name' not found locally. Attempting to pull from Docker Hub..."
    docker pull $image_name
    if [ $? -ne 0 ]; then
        echo "Error: Failed to pull image '$image_name' from Docker Hub."
        exit 1
    fi
fi

# Create the container with specified configuration
docker run -d \
```

```
--name $container_name \
-v ./web_santicolle:/var/www/html/ \
-p 8090:80 \
--cap-add NET_ADMIN \
--user root \
--restart always \
--network red_web \
--ip 192.168.10.10 \
--hostname servidor-web \
$image_name
```



```
santi@santi-PC:~/Escritorio/SWAP/P1
~/E/SWAP/P1 docker network rm red_web
docker network rm red_servicios
red_web
red_servicios

~/E/SWAP/P1 ./scripts/redes.sh red_web 192.168.10.0/24
c499f4246c2dd9062b5cb368d8fe91ac40d350bc1bf709c66607eb7442d274c3

~/E/SWAP/P1 ./scripts/redes.sh red_servicios 192.168.20.0/24
824565ae09e5db4ef357448912ee9180f9ef84f29ed4f08a4d9cf68500907832

~/E/SWAP/P1 docker network ls
NETWORK ID      NAME                DRIVER  SCOPE
6f729d7949b3    bridge             bridge  local
8cde8947cf16    host              host    local
5ea3c3a06d80    none             null    local
824565ae09e5    red_servicios     bridge  local
c499f4246c2d    red_web          bridge  local

~/E/SWAP/P1
```

Name	Image	Status	CPU (%)	Port(s)
 prueba 7f7c59de7a07	santicolle- 	Running	0.01%	8090:80

```
santi@santi-PC:~/Escritorio/SWAP/P1
~/E/SW/P1 ./scripts/contenedores.sh prueba santicolle-apache-image:p1
7f7c59de7a07c2951a7ff48be64e7a0b5b4d7327be8326d1b240c0bd68c72f1c

~/E/SWAP/P1
```

- Análisis propuesta IA

He tenido muchos problemas con el reconocimiento de los comandos en el Dockerfile pero la IA tampoco sabía muy bien cómo arreglarlos, en un momento había puesto dobles comillas dentro de otras dobles comillas y no me estaba ejecutando bien las sentencias, al final me he dado cuenta del fallo y le he puesto comillas simples pero por mi cuenta, no porque me lo dijera la IA. En cuanto a comandos, la mayoría relacionados con cron si han sido generados y tanto Gemini como ChatGPT me daban soluciones parecidas, por lo que supongo que no estará del todo mal. El problema es que no puedo comprobar si se ejecuta o no diariamente.

Los scripts de bash que me han generado para la creación de redes y contenedores funcionan pero para los contenedores, si es una imagen que no está en Docker Hub, debemos hacer el build previamente, y tiene un problema ya que añade una red web y un puerto predeterminados que tendrían que ser modificados para varios contenedores.