

## Ejercicio 1:

La empresa para la que trabajas necesita desplegar la aplicación PHP proporcionada, utilizando contenedores Docker. Para ello tienes que:

1. Crear un Dockerfile que:
  - a. Parta de la imagen php:8.2-apache.
  - b. Establezca el directorio de trabajo adecuado. Piensa como podrías saberlo.
  - c. Copie todos los archivos de la aplicación al contenedor
  - d. Exponga el puerto 80
  - e. Configure correctamente el arranque del servidor Apache. El comando adecuado es "apache2-foreground".
2. Construir una imagen llamada "miUsuario/miniapp-php:1.0", donde miUsuario será el propio de cada alumno.
3. Ejecutar un contenedor que:
  - a. Mapee el puerto 8080 del host al puerto 80 del contenedor
  - b. Se ejecute en segundo plano
  - c. Tenga nombre contenedor-miniapp
4. Comprobar el funcionamiento accediendo a:
  - a. <http://localhost:8080>
5. Y verificar que:
  - a. Se muestra la página principal.
  - b. Se puede navegar a la página de saludo.
  - c. Se aplica correctamente el CSS.

## Ejercicio 2:

La empresa para la que trabajas está desarrollando una pequeña herramienta interna en C, para realizar pruebas de entorno en distintos servidores. El objetivo de esta herramienta es:

1. Verificar que el compilador funciona correctamente.
2. Comprobar la ejecución interactiva por consola.
3. Leer archivos de configuración.
4. Validar que el entorno es consistente en todos los equipos.

Para evitar problemas derivados de diferencias entre sistemas (versiones de gcc, librerías, configuraciones...), la empresa ha decidido distribuir la aplicación dentro de un contenedor Docker, garantizando así que se ejecute siempre en el mismo entorno. Tu tarea como técnico/a de despliegue es crear la imagen Docker que compile y ejecute correctamente la aplicación proporcionada.

Se trata de una aplicación de consola interactiva que muestra un menú con varias opciones:

1. Mostrar la fecha y hora actual del sistema.
2. Generar un número aleatorio.
3. Leer y mostrar el contenido de un archivo de configuración (config/app.conf).
4. Salir del programa.

La aplicación:

1. Está desarrollada de forma modular (varios .c y .h).
2. Se compila usando un Makefile.
3. Depende de la correcta ubicación del archivo config/app.conf.

Tienes que crear una imagen que:

1. Compile correctamente la aplicación durante la fase de construcción.
2. Optimice la generación de capas.
3. Ejecute automáticamente el programa al iniciar el contenedor.
4. Permita la interacción desde consola.

Para ello, debes tener en cuenta que:

1. Se debe partir de la imagen gcc:13, que marca que el directorio de trabajo sea /usr/src/app.
2. Ten en cuenta que la aplicación necesita el fichero config/app.conf únicamente en tiempo de ejecución y que la compilación depende de los archivos fuente (.c y .h) y del Makefile.
3. Reflexiona sobre el orden en el que deben incorporarse los distintos elementos al contenedor durante el proceso de construcción.
4. El comando para compilar es make-
5. Para arrancar la aplicación solo hay que ejecutar "./app". app es generado a partir del Makefile cuando se compila.