

Apellidos: \_\_\_\_\_

Grupo: \_\_\_\_\_

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_

## Despliegue de Aplicaciones Web

### UD5: Docker – Despliegue mediante Imágenes y Contenedores

#### Tabla de respuestas:

1	2	3	4	5
D	C	B	A	B

6	7	8	9	10
A	B	A	C	D

11	12	13	14	15
D	D	D	A	A

16	17	18	19	20
B	B	A	A	D

Aciertos: \_\_\_\_\_

Errores: \_\_\_\_\_

En blanco: \_\_\_\_\_

#### Instrucciones del examen:

- Se deben responder a las preguntas en la **tabla de respuestas** dada.
- Las preguntas que se dejen en blanco deben ser marcadas con una 'X'.
- Todas las preguntas tienen la misma puntuación.
- Solamente hay una respuesta correcta en cada pregunta.
- Cada respuesta correcta suma 0,5.
- Cada respuesta incorrecta resta 0,5/3 (0,166 aproximadamente).
- Una respuesta en blanco ni suma ni resta.
- No se permite tener en posesión ningún aparato electrónico durante el examen. Responder con bolígrafo azul o negro. Minimizar el uso de tipex.
- El intento de copiar o fraude conllevará a la retirada del examen y a suspender con un 0 la prueba y obtener un 0 de actitud en la evaluación.

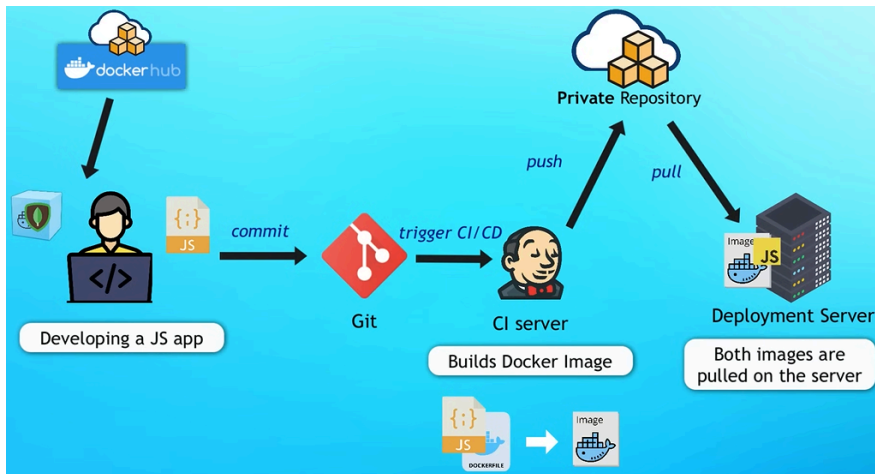
1. Es una desventaja de Docker-compose:
  - a) Requiere imágenes ya construidas, no es posible hacer build de un dockerfile con docker-compose.
  - b) No es posible controlar el orden de arranque de los contenedores.
  - c) No es posible tener datos persistentes; al hacer docker-compose down se borra todo.
  - d) **La gestión de una aplicación web que emplea miles de contenedores se hace inviable mediante un solo fichero docker-compose.**
2. ¿Que es Docker? ¿Cual es su finalidad?
  - a) Programa para la gestión portuaria marítima.
  - b) Un servidor que integra servicios de aplicaciones web, bases de datos, gestor de bases de datos, etc.
  - c) **Software de virtualización que facilita el desarrollo y el despliegue de aplicaciones.**
  - d) Sistema de backup para maquinas virtuales y repositorio de aplicaciones web.
3. ¿Cual es la diferencia entre una imagen y un contenedor de Docker?
  - a) A diferencia de los contenedores, una imagen no puede tener el entorno de desarrollo, solo el código de la aplicación a ejecutar.
  - b) **Un contenedor es una instancia en ejecución de una imagen.**
  - c) La imagen define los servicios a ejecutar, mientras que el contenedor define las variables de entorno y directorios necesarios.
  - d) Ninguna de las anteriores.
4. Si se ejecuta el siguiente comando vía consola:  
`docker run -d nginx:1.27`
  - a) **Arranca un contenedor en modo “detach”; la consola sigue funcional.**
  - b) Arranca un contenedor accesible vía la dirección IP 192.168.1.27.
  - c) Unicamente se descarga una imagen que contiene el servidor web Nginx.
  - d) Arranca un contenedor de servidor web Apache.
5. ¿Que se debe añadir o modificar a la instrucción definida en la pregunta anterior para poder acceder desde la maquina Host al contenedor?
  - a) `docker run -d -p nginx:80`
  - b) **`docker run -d nginx:1.27 -p 9000:80`**
  - c) `docker pull nginx:1.27 -p 9000:80`
  - d) `docker run -d nginx:1.27 -ip 172.0.0.1: 172.0.0.2`

6. Observa la siguiente imagen y completa los espacios, seleccionando una opción que resulte en un contenedor en marcha:



- a) **1: Build 2: Run.**  
b) 1: Pull 2: Run.  
c) 1: Run 2: Build.  
d) 1: Build 2: Start.
7. Revisa el siguiente código y determina la opción correcta:
- ```
FROM node:19-alpine
COPY package.json /app/
COPY src /app/
WORKDIR /app
RUN npm install
CMD ["node", "server.js"]
```
- a) Representa un Docker-compose. RUN pone en marcha el contenedor.  
**b) Representa un Dockerfile. COPY se emplea para copiar ficheros del host.**  
c) Representa un Dockerfile. Puede haber múltiples CMD.  
d) Representa un Docker-compose. El contenedor tendrá un servidor con distribución Alpine.
8. Continuando con el código de la pregunta 7, selecciona la opción correcta:
- a) **FROM indica una imagen base sobre la que se apilará la aplicación.**  
b) Todas las instrucciones declaradas se ejecutaran sobre el contenedor.  
c) Excepto CMD, todas las instrucciones se ejecutan simultáneamente.  
d) Todas las anteriores.

9. Observa el siguiente esquema y selecciona la opción correcta:



- a) El desarrollador no puede utilizar las imágenes con su aplicación, dado que solo se construyen en el servidor de CI (Integración continua).
  - b) El repositorio, el servidor de CI (Integración continua) y el pull del servidor son procesos que se deben realizar manualmente.
  - c) **Cuando se completa el ciclo, en el servidor se desplegarán contenedores con los mismos servicios y programas que los que ha utilizado y desarrollado el programador.**
  - d) Representa el ciclo de vida de un contenedor de Docker, desde su creación por parte del desarrollador hasta que se pasa a un servidor.
10. Selecciona la opción correcta de Docker-compose:
- a) Define qué servicios debe contener cada imagen y qué debe ejecutarse para arrancar cada contenedor.
  - b) Es imprescindible para poder arrancar contenedores.
  - c) Reemplaza Dockerfile para crear contenedores.
  - d) **Define instrucciones sobre cómo deben arrancar múltiples contenedores y trabajar en conjunto.**
11. Las siglas de YAML, en el ámbito de Docker, significa:
- a) Your Application Markup Language
  - b) Yet Another Machine Language
  - c) Yesterday Amazing Markup Language
  - d) **Yet Another Markup Language**

**12. Revisa el siguiente código y responde:**

```
services:
  mongo:
    image: mongo
    ports:
      - 27017:27017
    environment:
      - MONGO_INITDB_ROOT_USERNAME=admin
      - MONGO_INITDB_ROOT_PASSWORD=password
  mongo-express:
    image: mongo-express
    ports:
      - 8080:8081
    environment:
      - ME_CONFIG_MONGODB_ADMINUSERNAME=admin
      - ME_CONFIG_MONGODB_ADMINPASSWORD=password
      - ME_CONFIG_MONGODB_SERVER=mongo
```

- a) Es un Dockerfile.
- b) Es un fichero .env donde se definen variables de entorno.
- c) Es un fichero de configuración para arrancar un contenedor.
- d) Es un fichero .yaml o .yml.**

**13. Continuando con el código de la pregunta 12; ¿Qué sucede si se ejecuta un comando *docker-compose up* sobre dicho código? selecciona la opción correcta:**

- a) Se descargan las imágenes de mongo y mongo-express y arrancan en un único contenedor que contiene los servicios de mongo y mongo-express para trabajar en conjunto.
- b) Se construyen las imágenes de mongo y mongo-express y arrancan en dos contenedores aislados que contienen los servicios de mongo y mongo-express.
- c) Se descargan y se combinan las imágenes de mongo y mongo-express y arrancan en un único contenedor que contiene los servicios de mongo y mongo-express.
- d) Se descargan las imágenes de mongo y mongo-express y arrancan en dos contenedores respectivos, cada uno independiente, pero trabajando sobre una misma red.**

**14. Continuando con el código de la pregunta 12; ¿Qué aspecto del código permite a la máquina Host acceder a los servicios de base de datos (mongo) y su gestor (mongo-express)?**

- a) La definición de ports.**

- b) Es automático; docker-compose genera una red a la que añade todos los contenedores levantados y la máquina Host.
- c) La indentación de los servicios y sus parámetros indican a qué nivel deben trabajar.
- d) La definición de las variables de entorno.

15. ¿Qué acción realiza un comando *docker-compose down*?

- a) **Detiene y elimina los contenedores indicados por un fichero .yaml.**
- b) Elimina de la máquina Host las imágenes indicadas en el fichero .yaml.
- c) Es equivalente a realizar un *docker-compose stop*.
- d) Detiene los contenedores indicados por un fichero .yaml..

16. Si al código presentado en la pregunta 12, se añade la siguiente modificación ¿Que se consigue?:

```
services:
  mongo:
    image: mongo
    ports:
      - 27017:27017
    environment:
      - MONGO_INITDB_ROOT_USERNAME=admin
      - MONGO_INITDB_ROOT_PASSWORD=password
    volumes:
      - db-data:/data/db
(...)
volumes:
  db-data:
    driver: local
```

- a) Persistencia de datos a nivel de contenedor; Docker crea un contenedor aparte específico para guardar datos de otros contenedores.
- b) **Persistencia de datos a nivel local; se enlaza una unidad de la máquina host con un contenedor y los datos almacenados en este quedan guardados en dicho espacio.**
- c) Persistencia de datos a nivel de contenedor; se crea una unidad dentro de un contenedor y Docker guardará los datos del mismo aun cuando se detenga.
- d) Ninguna de las anteriores; no puede haber persistencia de datos con contenedores.

17.: Sobre el código anterior, pregunta 16, ¿Bajo que nombre reconoce Docker el volumen enlazado?

- a) mongo.
- b) db-data.**
- c) local.
- d) /data/db.

18.: Analiza las dos siguientes instrucciones y responde la opción correcta:

Docker network create mongo-network

```
docker run -p 27017:27017 -d  
-e MONGO_INITDB_ROOT_USERNAME=admin  
-e MONGO_INITDB_ROOT_PASSWORD=password  
-e MONGO_INITDB_DATABASE=user-account  
--name mongo --net mongo-network  
mongo
```

- a) El contenedor mongo estará accesible al resto de contenedores conectados a la red mongo-network, o vía puerto 27017.**
- b) El puerto 27017 no estará disponible para otros contenedores fuera de la red mongo-network.
- c) Las variables de entorno son siempre opcionales y se pueden omitir.
- d) El contenedor mongo será accesible desde cualquier red de Docker sin restricciones.

19. ¿Que diferencia los contenedores de Docker de una maquina virtual de VirtualBox?

- a) Los contenedores Docker comparten el mismo kernel del Sistema Operativo (SO) host. Docker no virtualiza un SO completamente.**
- b) Docker crea máquinas virtuales específicas para ejecutar aplicaciones de forma aislada. Las máquinas virtuales de VirtualBox son de uso general.
- c) Docker virtualiza toda la infraestructura, simulando un hardware diferente del host.
- d) Cada contenedor de Docker tiene su propio SO completo, incluyendo su propio kernel.

20. ¿Qué es Kubernetes?

- a) Kubernetes es un sistema operativo.
- b) Kubernetes es una base de datos relacional utilizada para almacenar grandes volúmenes de datos de contenedores.
- c) Kubernetes es el repositorio de imágenes público de Docker.
- d) Kubernetes ofrece un entorno de administración centrado en contenedores.**