

**Daniel Gutiérrez Navío**

---

# Introducción a Contendores

## Práctica 2: Persistencia y Docker compose

# Introducción

Durante esta práctica se reforzarán los conceptos estudiados en la lección, en particular Persistencia y Docker Compose, mediante la realización de 4 ejercicios, 2 de Persistencia y 2 de Docker Compose.

La entrega de la práctica deberá ser realizada en un archivo comprimido en .zip o .tar.gz, con las siguientes características:

- El nombre de la carpeta debe ser `practica2_IC_<nombre>_<apellido1>`
- El Zip tiene que contener 4 carpetas:
  - `persistencia_1` contendrá la solución pedida en el ejercicio 1
  - `persistencia_2` contendrá la solución pedida en el ejercicio 2
  - `docker-compose_1` contendrá la solución pedida en el ejercicio 3
  - `docker-compose_2` contendrá la solución pedida en el ejercicio 4

La entrega se calificará con un máximo de 10 puntos, correspondiendo 2.5 puntos a cada ejercicio correcto

## Persistencia

### 1: Migración de persistencia en base de datos

Instalar un contenedor de la imagen 'postgres', con un volumen con nombre 'db-persistence', usando la versión 9.6.1 de Postgres. Tras ello, eliminar el contenedor y levantar un nuevo postgres, pero con versión 9.6.2 usando el volumen anteriormente creado.

La entrega de este ejercicio debe ser un pdf compuesto por las capturas de:

- Instalación de ambos postgres con log
- Salida de 'docker volume inspect db-persistence'
- Logs de ambos postgres indicando que se ha iniciado correctamente, con 'docker logs <container>'

### 2: Realizar un bind mount para modificar código dinámicamente

Instalar un contenedor con la imagen de la Práctica 1, usando un bind mount para ser capaz de cambiar el texto que salga por pantalla por un mensaje diciendo el nombre del alumno.

La entrega de este ejercicio debe ser compuesta por las capturas de:

- Lanzamiento del docker
- Salida de 'docker volume inspect <container>'

- Captura de imagen del display de pantalla del servicio web

# Docker Compose

## 3: Docker Compose, build image y Persistencia

Realizar un Docker Compose que sea capaz de levantar el servicio de Node realizado en la práctica anterior, construyendo el mismo con el propio Docker Compose, no usando una imagen ya existente. Este servicio deberá incluir un Bind Mount, que permita alterar el contenido del archivo index.js.

El contenido de este ejercicio deberá ser la carpeta conteniendo el Dockerfile, el código de node, así como el docker-compose.yml.

## 4: Realizar un Docker Compose de Drupal y Postgres.

En esta última actividad trabajaremos en preparar un servidor de Drupal y Postgres orquestados dentro de Docker Compose. Este Docker Compose tendrá las siguientes características adicionales:

- El servicio debe exponerse en el puerto 32400 del hosts
- La base de Postgres deberá tener un volumen llamado "drupal-db"
- Ambos deberán estar dentro de una red llamada "drupal-network"

Como ayuda, se proporcionan los comandos de Docker para levantar tanto Drupal como Postgresql. Para mas información, se recomienda buscar las imágenes oficiales en Dockerhub

**Drupal:**

```
docker run --name some-drupal --network some-network -d \  
-v drupal-modules:/var/www/html/modules \  
-v drupal-profiles:/var/www/html/profiles \  
-v drupal-sites:/var/www/html/sites \  
-v drupal-themes:/var/www/html/themes \  
drupal
```

**Postgresql:**

```
docker run -d -v pgdata:/var/lib/postgresql/data --name my-postgres -e  
POSTGRES_PASSWORD=mysecretpassword -d postgres
```

El contenido de este ejercicio deberá ser una carpeta llamada docker-compose\_2, con un docker-compose.yml conteniendo la solución al ejercicio.