

## 4 - PERSISTENCIA Y DOCKER COMPOSE

### **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"

El contenido de este ejercicio deberá ser una carpeta llamada `docker-compose_2`, con un `docker-compose.yml` conteniendo la solución al ejercicio.

Se crea una base de datos postgres según lo solicitado, con red "drupal-network" y data en volumen drupal-db

```
version: '3.9'
```

```
services:
```

```
  postgres:
```

```
    image: postgres
```

```
    restart: always
```

```
    environment:
```

```
      POSTGRES_DB: postgresdb
```

```
      POSTGRES_USER: postgresuser
```

```
      POSTGRES_PASSWORD: postgrespassword
```

```
    volumes:
```

```
      - drupal-db:/var/lib/postgresql/data
```

```
    ports:
```

```
      - 5432:5432
```

```
    networks:
```

```
      - drupal-network
```

Se vinculan los volúmenes que participan de acuerdo al script de ejemplo. Se asocia a la red “drupal-network”. Finalmente, se expone en puerto 32400 según lo solicitado y como adicional se genera dependencia con la instancia de postgres.

```
drupal:
  image: drupal
  restart: always
  ports:
    - 32400:80
  volumes:
    - drupal-modules:/var/www/html/modules
    - drupal-profiles:/var/www/html/profiles
    - drupal-sites:/var/www/html/sites
    - drupal-themes:/var/www/html/themes
  depends_on:
    - postgres
  networks:
    - drupal-network
```

Al final, se declaran los volúmenes que participan y se configura el driver “bridge” para el network solicitado

```
- volumes:
  drupal-db:
  drupal-modules:
  drupal-profiles:
  drupal-sites:
- drupal-themes:

- networks:
- drupal-network:
-   driver: bridge
```