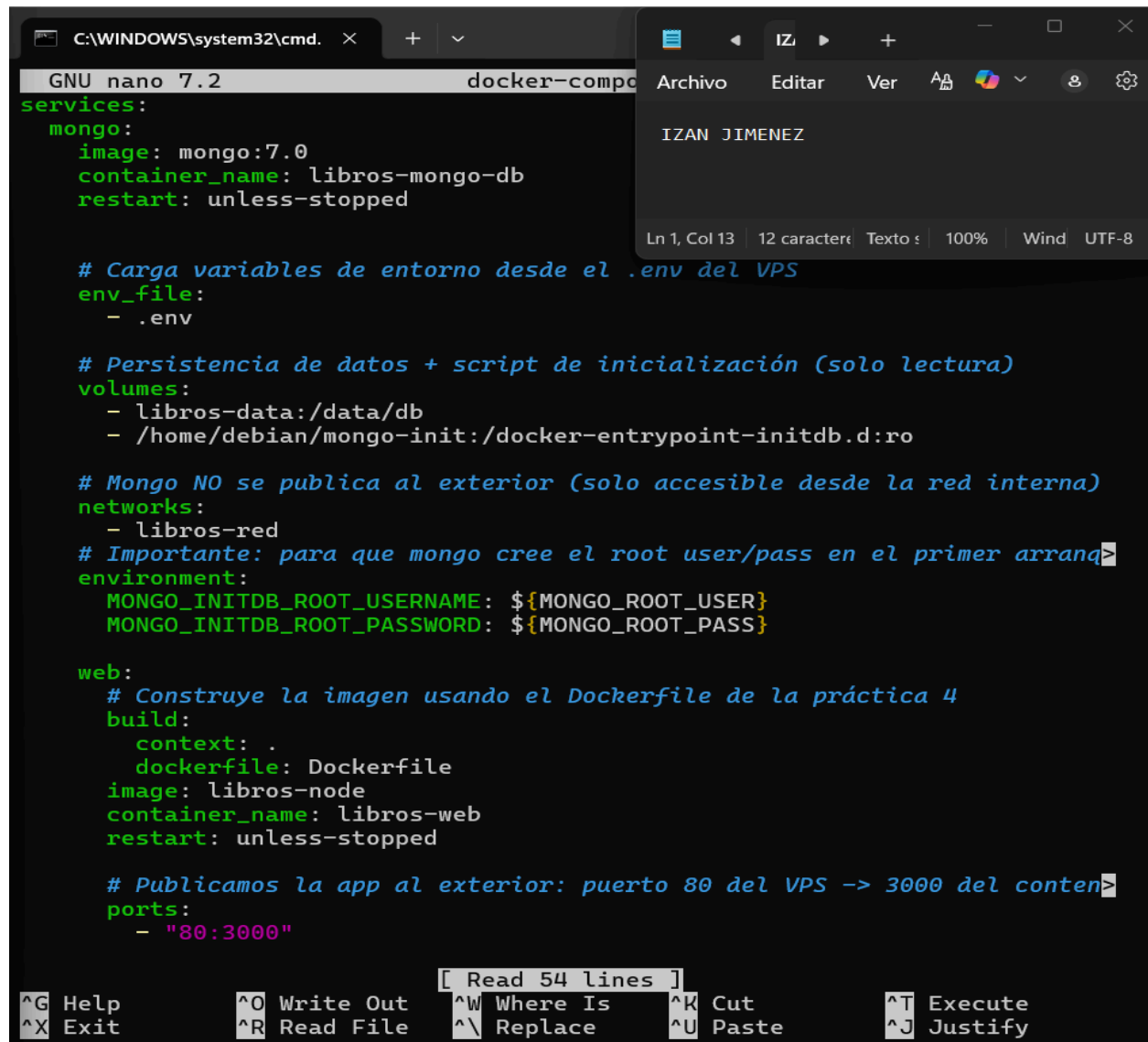


Práctica 5. Orquestación de una aplicación Node.js con Docker Compose

Captura del archivo docker-compose.yml con explicación de cada sección:

Las explicaciones van en la captura



```
GNU nano 7.2 docker-compo
services:
  mongo:
    image: mongo:7.0
    container_name: libros-mongo-db
    restart: unless-stopped

    # Carga variables de entorno desde el .env del VPS
    env_file:
      - .env

    # Persistencia de datos + script de inicialización (solo lectura)
    volumes:
      - libros-data:/data/db
      - /home/debian/mongo-init:/docker-entrypoint-initdb.d:ro

    # Mongo NO se publica al exterior (solo accesible desde la red interna)
    networks:
      - libros-red
    # Importante: para que mongo cree el root user/pass en el primer arranque
    environment:
      MONGO_INITDB_ROOT_USERNAME: ${MONGO_ROOT_USER}
      MONGO_INITDB_ROOT_PASSWORD: ${MONGO_ROOT_PASS}

  web:
    # Construye la imagen usando el Dockerfile de la práctica 4
    build:
      context: .
      dockerfile: Dockerfile
    image: libros-node
    container_name: libros-web
    restart: unless-stopped

    # Publicamos la app al exterior: puerto 80 del VPS -> 3000 del contenedor
    ports:
      - "80:3000"

[ Read 54 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut
^X Exit      ^R Read File  ^\ Replace   ^U Paste
^T Execute   ^J Justify
```

```
# Variables de entorno de la app (MONGO_URI, PORT, SESSION_SECRET, et>
env_file:
  - .env

networks:
  - libros-red

# Arranca después de mongo (ojo: no espera "ready", solo orden)
depends_on:
  - mongo

volumes:
  libros-data:

networks:
  libros-red:
    name: libros-red
```

^G Help ^X Exit ^O Write Out ^R Read File ^W Where Is ^\ Replace ^K Cut ^U Paste ^T Execute ^J Justify

Captura del comando docker compose up -d ejecutándose correctamente:

Este comando crea y arranca la aplicación

```
C:\WINDOWS\system32\cmd. x + v
root@vps-d0ce1e12:/home/debian/libros# docker compose down
root@vps-d0ce1e12:/home/debian/libros# docker compose up -d
WARN[0000] a network with name libros-red exists but was not c
Set 'external: true' to use an existing network
network libros-red was found but has incorrect label com.docke
root@vps-d0ce1e12:/home/debian/libros# docker rm -f libros-web
libros-web
root@vps-d0ce1e12:/home/debian/libros# docker network rm libro
libros-red
root@vps-d0ce1e12:/home/debian/libros# docker compose up -d
[+] Running 4/4
 ✓ Network libros-red          Created
 ✓ Volume libros_libros-data  Created
 ✓ Container libros-mongo-db   Started
 ✓ Container libros-web        Started
root@vps-d0ce1e12:/home/debian/libros# SOY IZAN J|
```

Captura de docker compose ps mostrando los servicios activos:

Aquí se muestran los contenedores de servicios activos

```
C:\WINDOWS\system32\cmd. x + v
root@vps-d8ce1e12:/home/debian/libros# docker compose ps
NAME                IMAGE                COMMAND              SERVICE    CREATED          STATUS          PORTS
libros-mongo-db      mongo:7.0            "docker-entrypoint.s..."  mongo      About a minute ago Up About a minute 27017/tcp
libros-web           libros-node          "docker-entrypoint.s..."  web        About a minute ago Up About a minute 0.0.0.0:80->3000/tcp, [::]:80->3000/tcp
root@vps-d8ce1e12:/home/debian/libros# SOY IZAN J
```

Captura de los logs donde se observe la inicialización de MongoDB:

Con el comando docker compose logs mongo podemos ver el registro de los logs de los contenedores y lo que sucede con ellos y lo que ha sucedido

```
C:\WINDOWS\system32\cmd. x + v
}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:01:57.136+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"","msg":"client metadata","attr":{"remote":"172.18.0.3:58372","client":"conn4","negotiatedCompressors":[],"doc":{"driver":{"name":"nodejs|Mongoose","version":"5.9.2|7.8.8"},"platform":"Node.js v20.20.0, LE","os":{"name":"linux","architecture":"x64","version":"6.1.0-42-cloud-amd64","type":"Linux"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:02:46.166+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037266,"ts_usec":166456,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 4, snapshot max: 4 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:03:46.177+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037426,"ts_usec":177651,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 5, snapshot max: 5 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:04:46.187+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037486,"ts_usec":187849,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 6, snapshot max: 6 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:05:46.192+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037546,"ts_usec":192696,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 7, snapshot max: 7 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:06:46.198+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037606,"ts_usec":198841,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 9, snapshot max: 9 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:07:46.213+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037666,"ts_usec":213730,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 11, snapshot max: 11 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:08:46.222+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037726,"ts_usec":222553,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 12, snapshot max: 12 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:09:46.228+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037786,"ts_usec":228172,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 13, snapshot max: 13 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
libros-mongo-db | {"t":{"$date":"2026-02-02T13:10:46.235+00:00"},"s":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger message", "attr":{"message":{"ts_sec":1770037846,"ts_usec":235819,"thread":"1:0x7fd24bed7600","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOINT_PROGRESS","category_id":6,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"saving checkpoint snapshot min: 14, snapshot max: 14 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 8"}}}}
root@vps-d8ce1e12:/home/debian/libros# SOY IZAN J
```

Captura de la aplicación funcionando en el navegador:

Mediante la URL del VPS podemos acceder a nuestro contenedor de Libros

