

# Práctica de Ruby on Rails

DESARROLLO DE APLICACIONES WEB

SANDRA RUIZ JIMÉNEZ

En esta práctica montaremos un entorno de desarrollo para el framework Ruby on Rails utilizando como base el docker-compose.yml de Bitnami que podemos encontrar en la siguiente URL: <https://hub.docker.com/r/bitnami/rails/>

Una vez hemos descargado el fichero docker-compose.yml asociado deberemos modificarlo de la siguiente manera:

- Modificar el puerto de la aplicación al que nos podremos conectar desde nuestro host para que sea el puerto 8001.
- Posteriormente deberemos iniciar la aplicación multicapa con docker-compose up.

Debes entregar los siguientes ficheros:

- El fichero docker-compose.yml modificado y explicando cada una de las líneas del fichero
- Un pantallazo donde se pueda ver la ejecución de la orden docker-compose.
- Un pantallazo donde se pueda apreciar que la aplicación base del framework está en ejecución a través de <http://127.0.0.1:8001> en vuestro navegador.

Haz un manual que explique el ejercicio paso por paso.

Se valorará presentación y comprensión de la explicación.

Creamos un directorio para la aplicación:

```
mkdir ~/myapp
cd ~/myapp
```

Descargamos el fichero docker-compose.yml con el siguiente comando:

```
$ curl -LO https://raw.githubusercontent.com/bitnami/containers/main/bitnami/rails/docker-compose.yml
```

Entramos en el fichero docker-compose.yml para editarlo:

```
docker-compose.yml
version: '2'  ← version de docker compose

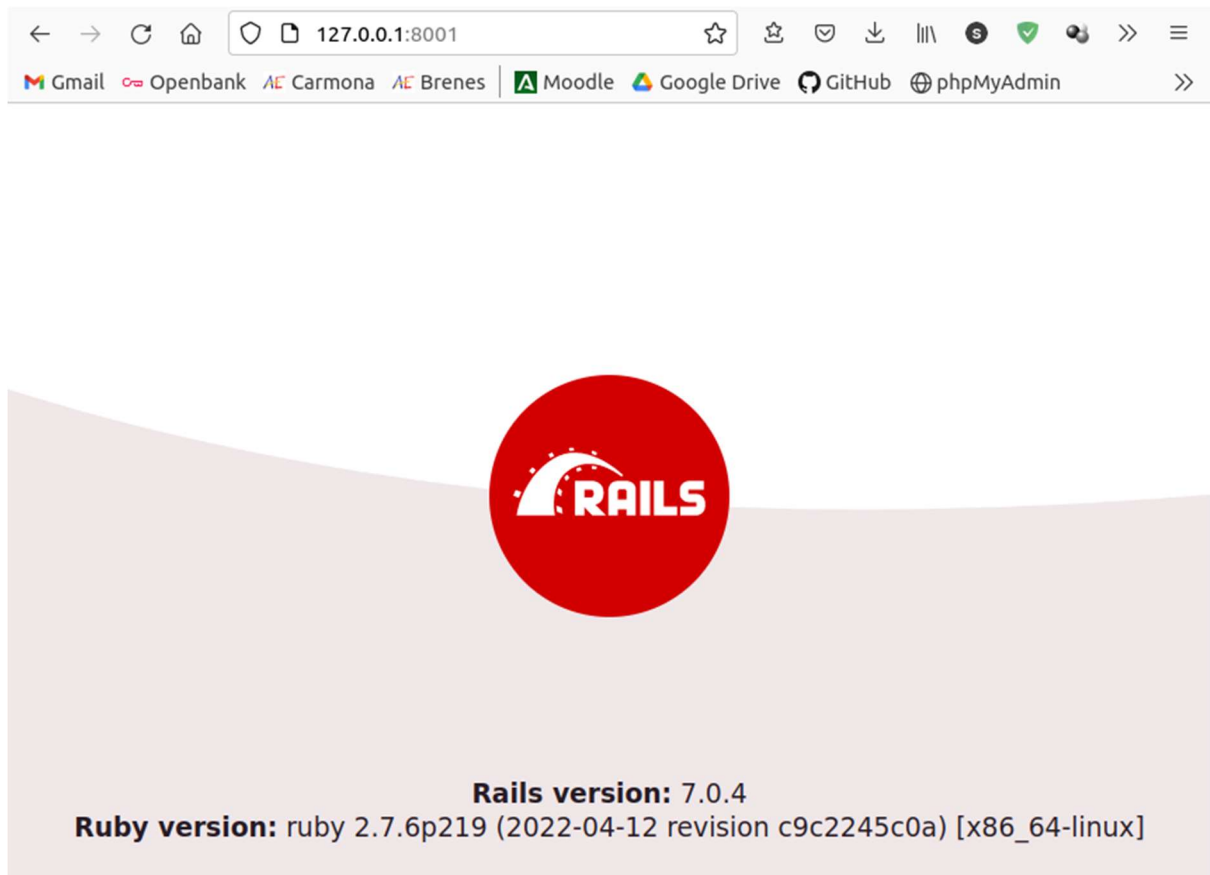
services:  ← configuración de todos los contenedores, en este caso, mariadb y myapp (rails)
  mariadb:
    image: docker.io/bitnami/mariadb:10.6  ← desde donde descargará la imagen y su versión
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes  ← permite contraseñas vacías en la bdd
  myapp:
    image: docker.io/bitnami/rails:7  ← desde donde descargará la imagen y su versión
    ports:
      - '8001:3000'  ← puerto de redireccionado: puerto por el que escucha rails
    environment:
      - DATABASE_HOST=mariadb  ← nombre del host de la bdd
      - DATABASE_NAME=my_app_development  ← nombre de la bdd
    volumes:
      - './my-project:/app'  ← volumen donde se persistirán los datos
    depends_on:
      - mariadb  ← este contenedor depende del contenedor mariadb
```

Ejecutamos docker-compose up (siempre donde esté el fichero docker-compose.yml) y en la primera vez se nos van a descargar las imágenes que están en el fichero .yml.

A la vez, este comando levanta los contenedores:

```
myapp_1 | rails 16:20:32.73 INFO ==> Configuring database name to my_app_development
myapp_1 | rails 16:20:32.73 INFO ==> Connecting to the database at mariadb (type: mysql)
mariadb_1 | 2022-11-06 16:20:32.3 [Warning] Aborted connection 3 to db: 'unconnected' user: 'unauthenticated' host: '172.19.0.3' (This connection closed normally without authentication)
myapp_1 | rails 16:20:32.74 INFO ==> Initializing database (db:prepare)
myapp_1 | Created database 'my_app_development'
myapp_1 | rails 16:20:35.84 INFO ==> Database was successfully initialized
myapp_1 | rails 16:20:35.86 INFO ==> ** Rails setup finished! **
myapp_1 | ==> Booting Puma
myapp_1 | ==> Rails 7.0.4 application starting in development
myapp_1 | ==> Run 'bin/rails server --help' for more startup options
myapp_1 | Puma starting in single mode...
myapp_1 | * Puma version: 5.6.5 (ruby 2.7.6-p219) ("Birdie's Version")
myapp_1 | * Min threads: 5
myapp_1 | * Max threads: 5
myapp_1 | * Environment: development
myapp_1 | * PID: 1
myapp_1 | * Listening on http://0.0.0.0:3000
myapp_1 | Use Ctrl-C to stop
```

En el navegador, al escribir 127.0.0.1:8001, vemos que accedemos a la aplicación Rails:



Para pararla, pulsamos Ctrl+C en el Terminal.