

# Práctica final seminario Docker

El proyecto consiste en un contador y expone las siguientes urls:

- /
- /incrementar
- /decrementar

Está desarrollado con un framework llamado Sinatra (desarrollado con el lenguaje Ruby) que se conecta con un servicio Redis. El proyecto consta de 3 archivos: config.ru, Gemfile y myapp.rb.

Ejercicio 1 - Construir un Dockerfile con las siguientes características:

- La imagen base tiene que ser **ruby:3.0**
- El contenido del proyecto debe ser copiado a un folder llamado "**myapp**" dentro de la imagen.
- En la misma carpeta donde se copiaron los archivos, se debe ejecutar el comando "**bundle install**".
- Se debe configurar la imagen (con una sentencia en el Dockerfile) de manera que cuando se inicialice un container, se ejecute el siguiente comando (en la misma carpeta donde se copiaron los archivos): **bundle exec rackup --host 0.0.0.0 -p 4567**

## Validación del ejercicio

La imagen construida es una aplicación Sinatra en el puerto **4567**. Para validar que está funcionando correctamente se debe ejecutar un contenedor (con el comando docker de línea de comando, sin utilizar de momento docker-compose) utilizando la imagen construida y realizando el mapeo de puerto correcto. Una vez ejecutada esta aplicación con un navegador acceder a **localhost:4567** y obtener un mensaje de error similar al siguiente:



**Redis::CannotConnectError at /**  
**Error connecting to Redis on 10.0.1.1:6380 (Redis::TimeoutError)**  
file: client.rb location: rescue in establish\_connection line: 398

## Ejercicio 2 - Construir un docker-compose

Construir un docker compose que construya la imagen del ejercicio 1 y agregue un servicio Redis. La imagen del servicio redis es **redis**. (Validar que el host en la línea 10 del archivo myapp.rb sea el correcto)

### Validación del ejercicio

Para validar el ejercicio realizar los siguientes tests:

1. Abrir la url: **localhost:4567/** y validar que muestra el valor del contador.
2. Abrir la url: **localhost:4567/incrementar** y validar el incremento del contador.
3. Abrir la url: **localhost:4567/decrementar** y validar el decremento del contador.

## Ejercicio 3 - Montar un volumen

Modificar los ejercicios anteriores para que el proyecto no sea copiado dentro de la imagen construida sino que el código fuente sea mapeado de una carpeta local.

### Validación del ejercicio

1. Construir y ejecutar los containers utilizando docker compose.
2. Abrir la url: **localhost:4567/** y ver que el mensaje sea: "Contador: x"
3. Sin detener los contenedores, modificar el archivo myapp.rb y cambiar la palabra "Contador" por "Cuenta".
4. Recargar la url: **localhost:4567/** y verificar que ahora el mensaje sea "Cuenta: x"

## Formato de entrega de los ejercicios

- Entregar los ejercicios en 3 carpetas separadas, 1 carpeta por cada ejercicio.
- La carpeta "ejercicio 1" debe contener el código del proyecto y el Dockerfile.
- La carpeta "ejercicio 2" debe contener el código del proyecto, el Dockerfile y el docker compose.
- La carpeta "ejercicio 3" debe contener los mismos archivos que el ejercicio 2, pero modificados para cumplir con el ejercicio 3.