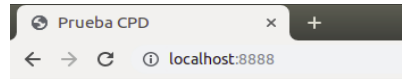


Práctica 2 - Contenedores Docker

1. Crear una imagen personalizada de docker con Apache que cuando se acceda a <http://localhost:8888> aparezca vuestro nombre (editando el index.html) (según apartado I).



Elena Cantero Molina

Dockerfile

```
FROM debian
MAINTAINER elenacm "eleniita2197@gmail.com"

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf
/var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2


EXPOSE 80
ADD ["index.html", "/var/www/html/"]


ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```


index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Prueba CPD</title>
  </head>
  <body>
    <h1>Elena Cantero Molina</h1>
  </body>
</html>
```

A partir de los pasos explicados en el apartado II subir la imagen a hub.docker.com e indicar el nombre de la imagen creada.



 **elenacm / pruebapcd**

This repository does not have a description 

 Last pushed: 2 minutes ago

Tags

This repository contains 1 tag(s).

prueba1		 a minute ago
---------	---	---

[See all](#)

Full Description

Imagen creada de prueba

2. Contenido del fichero Dockerfile personalizado del apartado II y ficheros utilizados

versión 1

```
FROM alpine:3.8
RUN apk update
RUN apk add curl
RUN apk add vim
```

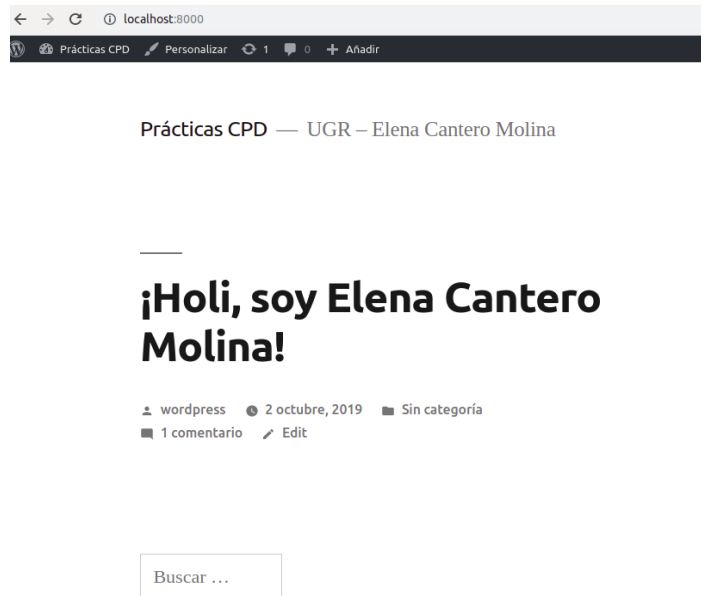
versión 2

```
FROM alpine:3.8
RUN apk update && apk add curl && apk add vim && apk add git
```

versión 3

```
FROM alpine:3.8
RUN apk update && apk add \
curl \
git \
vim
```

3. Según el apartado III, una vez desplegado el servidor Wordpress, editar la página principal para que aparezca el nombre del usuario y realizar una captura de pantalla.



4. Siguiendo los pasos del apartado IV, realice un programa en C o Python, que permita evaluar el tiempo de ejecución de un algoritmo (ej. el cálculo de las N primeras cifras de PI), y comprobar cómo podemos aumentar o reducir la CPU dedicada y por tanto el tiempo de ejecución. Prepare un contenedor con dicho experimento, súbalo a hub.docker.com e indique en el documento los pasos que realiza para el experimento y los tiempos de ejecución obtenidos.
- Realizar el programa para calcular las cifras de PI.

```
#include <stdio.h>
#include <time.h>

int main(void)
{
    register unsigned long long int i, j = 2, k = 1;
    long double pi = 2.0;

    clock_t init = clock();
    for (i = 0; i < 1000000000; ++i){
        pi *= (j / (double)k);

        if (!(i & 1))
            k += 2;
        else
            j += 2;
    }

    clock_t end = clock();

    double tiempo = ((float)(end - init))/CLOCKS_PER_SEC;

    printf("Tiempo : %f \n", tiempo);
    return 0;
}
```

- Crear el Dockerfile (usando los anteriores realizados) de manera que se pueda compilar el archivo .c

```
FROM alpine:3.8

COPY calculopi.c .

RUN apk update && apk add g++

RUN g++ calculopi.c -o calculopi
```


- Ejecutar `docker run -it --cpus=".5" calculopi ./calculopi`

Los tiempos tomados son los siguientes

cpus	tiempo
0.1	8.826916
0.3	8.372811
0.5	8.227486
0.7	8.128505
0.9	7.994394

- Subir la imagen a docker hub

 elenacm / calculopi

This repository does not have a description 

 Last pushed: 12 minutes ago

Tags

This repository contains 1 tag(s).

1.0



 12 minutes ago

[See all](#)