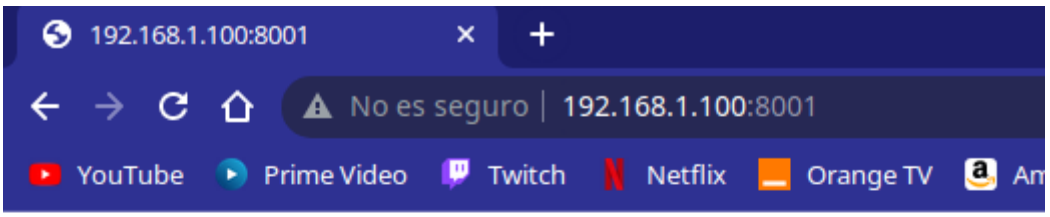
Documento de requisitos y diseño



Imagenes JPEG:

Ligera: Pesada:

Secuencia de 10 fotos JPG:

1 2 3 4 5 6 7 8 9 10

Animacion GIF:

No existente: enlace

Nota importante: enlace

Vídeo largo de 2GB: enlace

Texto .txt: enlace

Pacman: enlace

Snake: enlace

Descargar archivo de 165 KB: descargar

Scripts:

GET .py | POST .py | GET/POST .py

GET .php | POST .php

La página cargó en: 365ms

Rafael Domínguez Saez y Juan Francisco Flores Bocanegra

● Documento de requisitos

- El método GET funcionará perfectamente.
- El método POST funcionará perfectamente.
- El método OPTIONS funcionará perfectamente.
- En caso de no encontrarse la página deseada, se redirigirá al html e404.html.
- En caso de no disponer del método pedido, se redirigirá al html e400.html.
- El método GET podrá cargar elementos: html, jpg, jpeg, txt, gif, mpg, mpeg, doc, docx, pdf, py, php, ico, css, js, png y svg.
- El dueño del servidor decidirá en qué IP se ejecutará el servidor indicando el nombre de la interfaz.
- Se implementará la librería libconfuse para acceder a los datos escritos en server.conf.
- Se demonizará el proceso principal
- El proceso demonizado tendrá como directorio base www/ .
- El proceso demonizado tendrá los files STDIN, STDOUT y STDERR cerrados.
- El servidor dispondrá de hijos pre-creador que serán los encargados de aceptar las peticiones.
- El padre podrá recibir la señal SIGINT, con la que obligará a sus hijos terminar y cerrará el socket.
- Se dispondrá de un log (que se guarda en /var/log/syslog) y contendrá comentarios de la ejecución.
- La cabecera de las peticiones http contienen: Content-Type, Content-Length, Date, Last-Modified y Server.
- El programa podrá ejecutar los juegos Pacman y Snake.
- Se podrá descargar un fichero .pdf de la página.
- Los archivos estarán modularizados y en las carpetas correspondientes.
- El makefile generará 4 librerías (.a).

● Documento de diseño

Para la realización del diseño se optó por 5 ficheros principales:

- Main.c: Será el encargado de crear el socket, demonizar el proceso, obtener los valores dados en server.conf y de crear los hijos que serán los encargados de aceptar las distintas peticiones.
- Http.c: Leerá el contenido de la conexión, y gracias al método phr_parse_request obtendremos los datos importantes de la conexión, dependiendo del método pedido, llamaremos a las funciones GET, POST u OPTIONS, las cuales, serán las encargadas de enviar la información por a la conexión.
- Conf.c: Obtendrá los datos del fichero server.conf.
- Fileparser.c: Será la encargada de realizar acciones con los files (leer el contenido, obtener la extensión del fichero, etc).
- Ip.c: Dada una interfaz, devolverá la IP asignada a esa interfaz.
- Picohttpparser.c: Usado para la función phr_parse_request.