Universidad de Cádiz

Sistemas Distribuidos
Documentación ZeroMQ

Francisco José Pastor
Alejandro Chacón
Antonio Gambín
Iván Ruiz

7 de junio de 2015

Índice

1. Introducción.

1.1. Idea.

La idea de este proyecto es desarrollar un sistema cliente-servidor muy similar a FTP (File Transfer Protocol), en el cual el cliente conecta al servidor para realizar diferentes acciones, como puede ser listar un directorio, mover o copiar un archivo a otro directorio o eliminar un archivo. Nuestro programa estará dotado de estas acciones enumeradas anteriormente.

Aunque en un sistema FTP real el cliente puede realizar envíos de archivos al servidor o viceversa, en nuestro caso, el cliente está en fase de desarrollo y solamente podrá mover y copiar archivos de la carpeta del servidor a otra carpeta donde se encuentre este, es decir, si el cliente se encuentra en el mismo equipo que el servidor, la cual era nuestra idea original y es la versión que podemos encontrar en el directorio "local", podremos mover y/o copiar archivos a cualquier directorio del cliente, pero, si en su defecto el cliente se encuentra en otro equipo de la misma red o de una red externa (si se ha realizado la debida configuración y apertura de puertos necesarios), el cliente no podrá mover ni copiar archivos desde el servidor hacia él. En ninguno de los casos el cliente podrá enviar archivos al servidor.

1.2. Explicación del funcionamiento.

Para este proyecto hemos escogido la opción de transferencia de archivos, para la cual se ha realizado una conexión cliente-servidor a partir de la tecnología ZeroMQ, en la cual el cliente dispondrá de una serie de opciones dadas por el servidor, de las cuáles el cliente escribirá una serie de instrucciones y el servidor devolverá la solicitud propuesta. Esta conexión será punto a punto y el cliente se conectará al único servidor a través del único puerto que se habilitará.

En nuestra opción elegida, definimos una serie de instrucciones, para el correcto manejo de archivos en el servidor a través de las instrucciones dadas en el cliente, como son:

- Listar y visualizar los archivos que se encuentran en la misma carpeta que el servidor.
- Copiar los ficheros a cualquier otra carpeta del servidor, inclusive donde se encuentre el fichero de ejecución del servidor

- Mover los ficheros a cualquier otra carpeta del servidor, inclusive donde se encuentre el fichero de ejecución del servidor
- Eliminar los ficheros de la carpeta del servidor

2. Organización.

La realización del proyecto ha sido llevada a cabo por el equipo de desarrollo formado por los siguientes integrantes:

- Francisco José Pastor Aznar.
- Alejandro Chacón Peregrino.
- Antonio Rafael Gambín Iñigo.
- Iván Ruiz Valle.

2.1. Planificación.

En el siguiente diagrama de Gantt podemos observar las diferentes fases del desarrollo del proyecto y el tiempo que fue necesario para cada una de ellas:

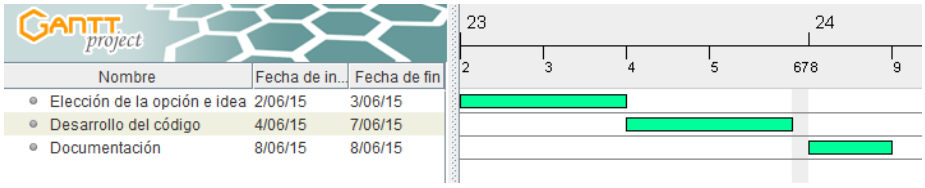

Figura 1: Planificación

2.2. Reparto de tareas.

En este proyecto el reparto de tareas ha sido equitativo entre todos los miembros del grupo.

2.3. Despliegue del proyecto.

El despliegue del proyecto se ha llevado a cabo mediante la herramienta de control de versiones *Github*, siendo esta la herramienta más conocida y empleada por desarrolladores actualmente, con el objetivo de favorecer una interacción directa entre los integrantes del grupo de trabajo y el proyecto, además de apoyar al software libre, ya que el código fuente está disponible para todos los públicos.

El proyecto está compuesto por una serie de elementos y estos se encuentran recogidos en una estructura jerárquica:

1. **Local:** Carpeta que contiene la versión del programa desarrollada para ejecutar en modo local.
   a) **Cliente:** Carpeta que contiene el código del cliente.
   b) **Servidor:** Carpeta que contiene el código del servidor.
2. **remoto:** Carpeta que contiene la versión del programa desarrollada para ejecutar en modo remoto.
   a) **Cliente:** Carpeta que contiene el código del cliente.
   b) **Servidor:** Carpeta que contiene el código del servidor.
3. **Documentacion.pdf:** Documentación realizada para el proyecto.
4. **README.md:** Fichero markdown que recoge la información básica del proyecto.

3. Variantes del proyecto.

Al ser el último proyecto de la asignatura, y con la correcta preparación para abordar el problema que nos concierne, realizamos la resolución del problema de una forma local, pero gracias a la versatilidad de la arquitectura y simpleza de la arquitectura efectuada, hicimos otra versión de carácter remota.

Es decir, permitimos la consulta completa de forma local, que era nuestro proyecto inicial, y además, conseguimos listar los ficheros de una máquina remota.

4. Ejecución del programa.

Para la ejecución, primero deberemos de instalar los paquetes necesarios:

- Si nos encontramos en un sistema basado en Debian, deberemos de tener instalado el gestor de módulos de aplicaciones de terceros en Python pip y en una terminal, escribir:

$ pip install pyzmq

- Si nos encontramos en otros sistemas, como puede ser Arch Linux, podremos instalarlo de una forma más sencilla como es:

$ yaourt python2-pyzmq

Una vez instalado las dependencias, pasamos a la ejecución de las instrucciones. Así, para la ejecución de la variante local, sería:

Una vez en el directorio correcto del servidor, ejecutaríamos el código del servidor:

$ python2.7 serverftp

Tras levantar el servidor, iremos al directorio donde se encuentre nuestro cliente y lo lanzaremos de la siguiente forma:

$ python2.7 clienteftp

y para la ejecución de la variante remota, sería:

$ python2.7 clienteftp[IP]
$ python2.7 serverftp

4.1. Pruebas.

Los resultados de las pruebas realizadas son los siguientes, estas pruebas han sido realizadas sobre la versión local del código:

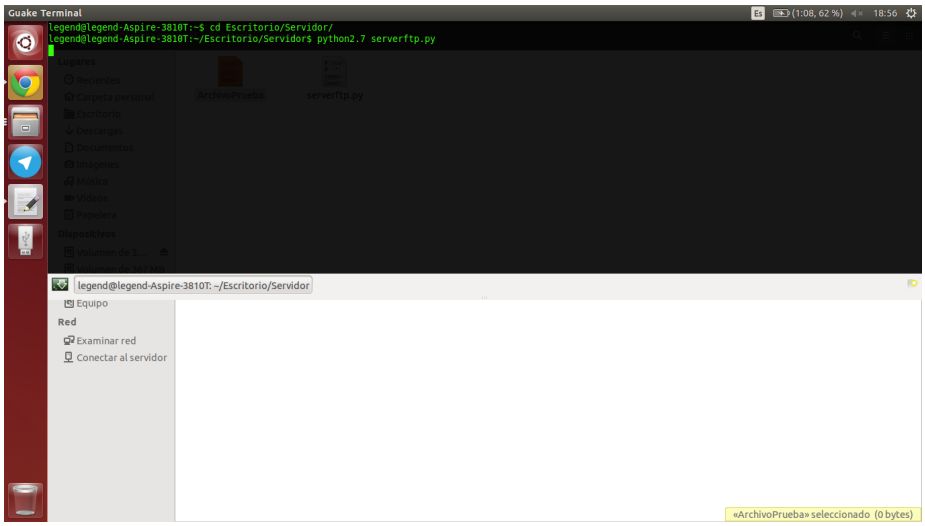

Figura 2: Ejecución servidor

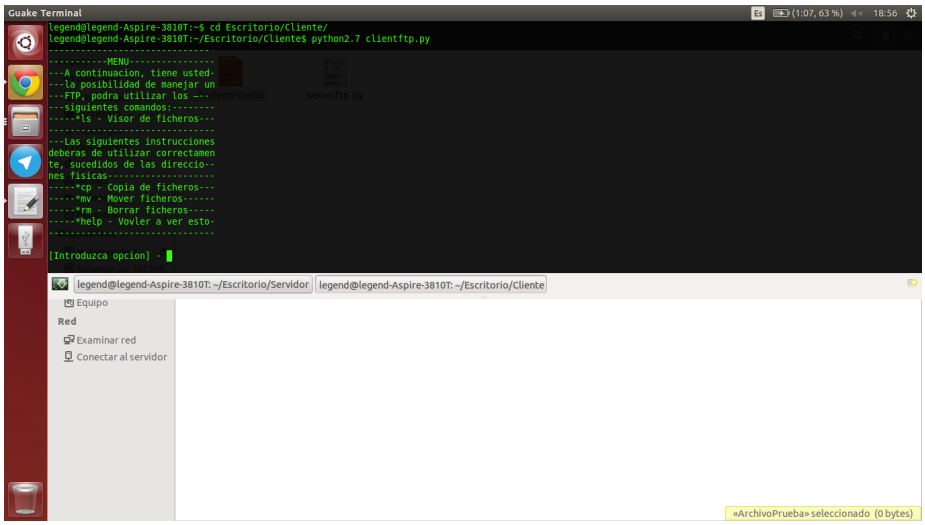

Figura 3: Ejecución cliente

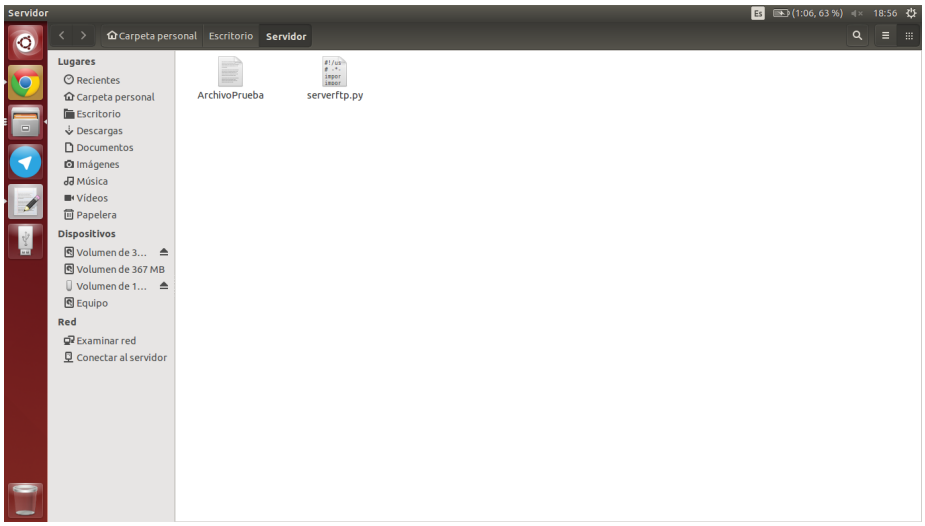

Figura 4: Comprobación carpeta servidor

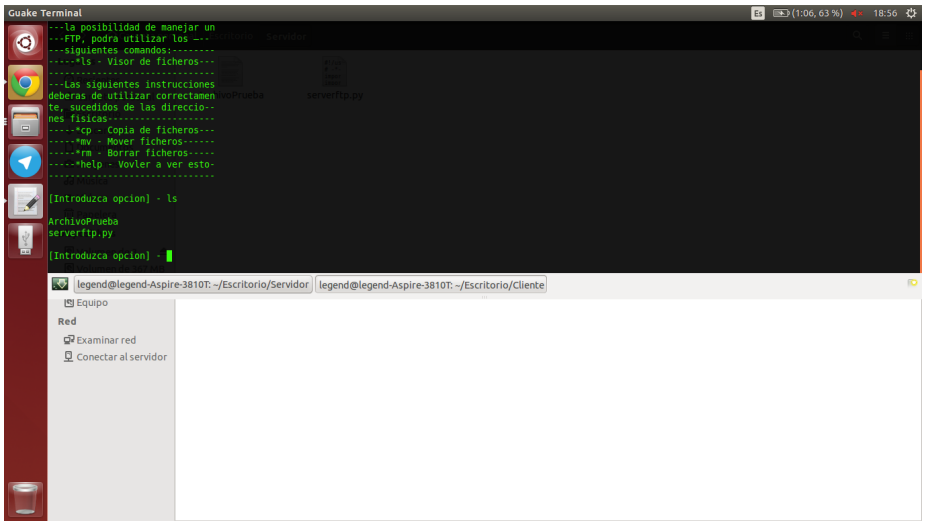

Figura 5: Prueba comando ls

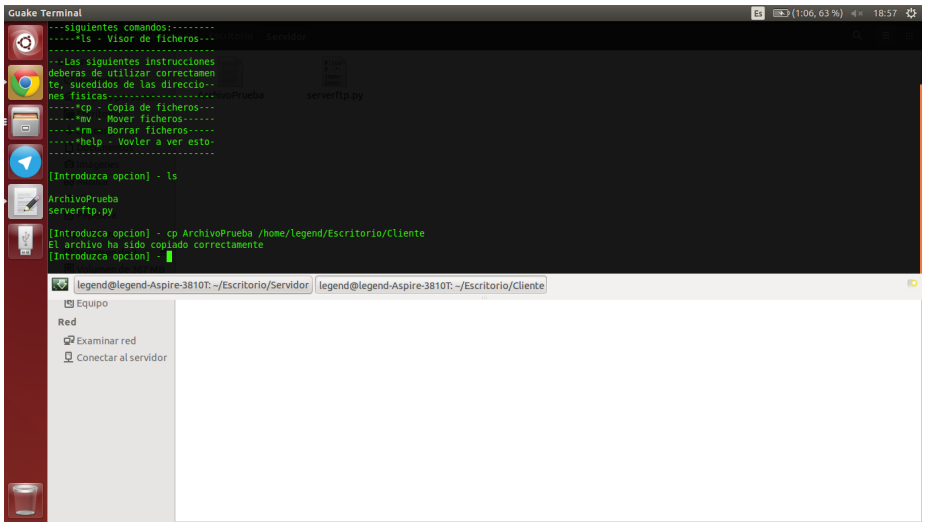

Figura 6: Prueba comando cp

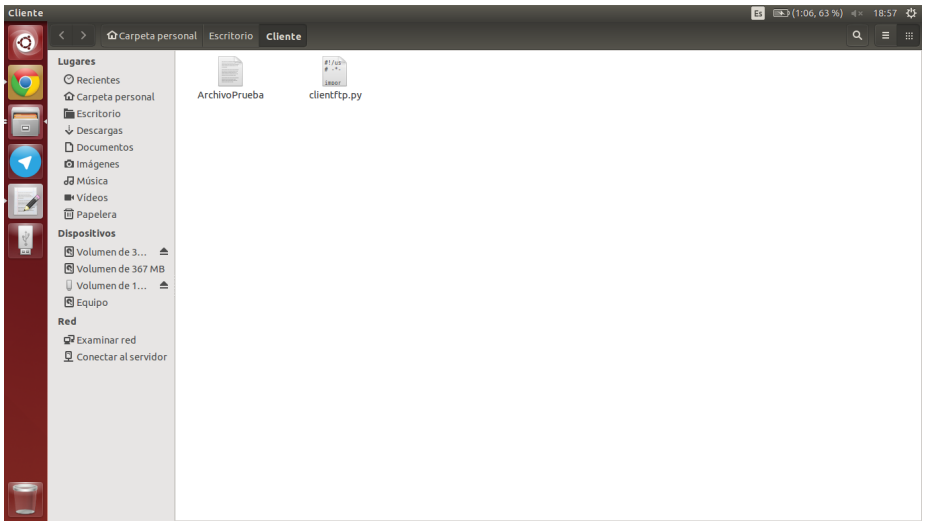

Figura 7: Comprobación de copia de archivo

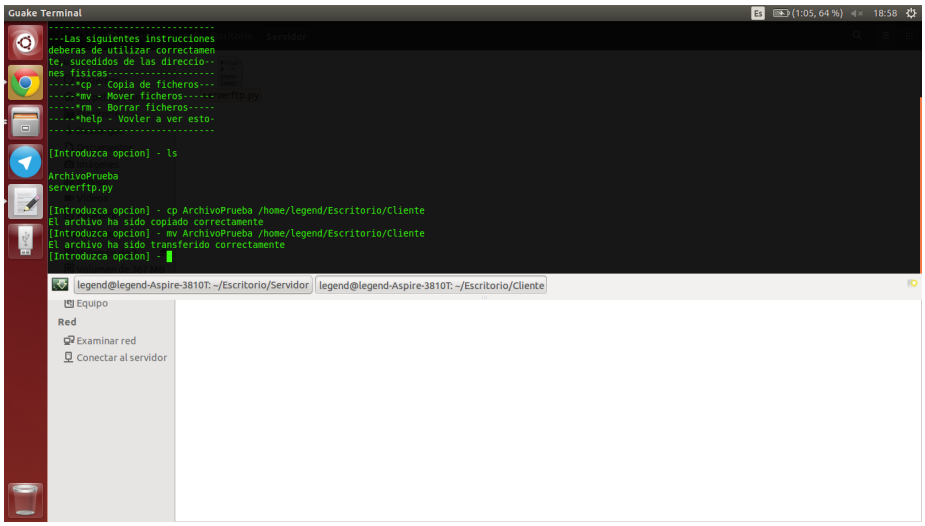

Figura 8: Prueba comando mv

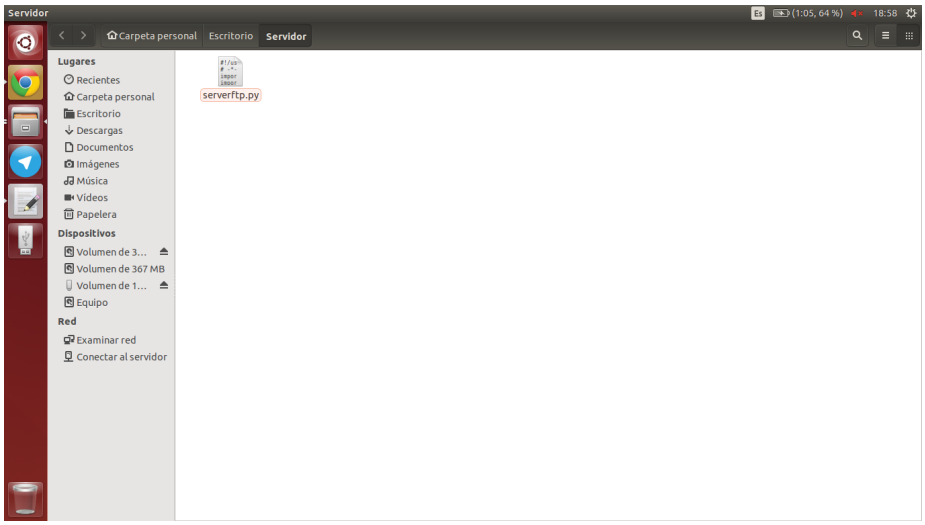

Figura 9: Comprobación de movimiento de archivo - 1

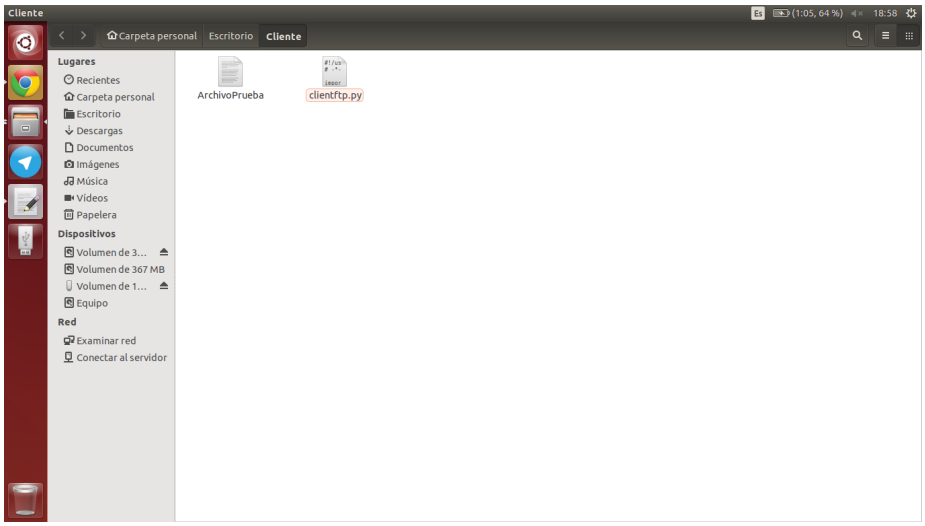

Figura 10: Comprobación de movimiento de archivo - 2

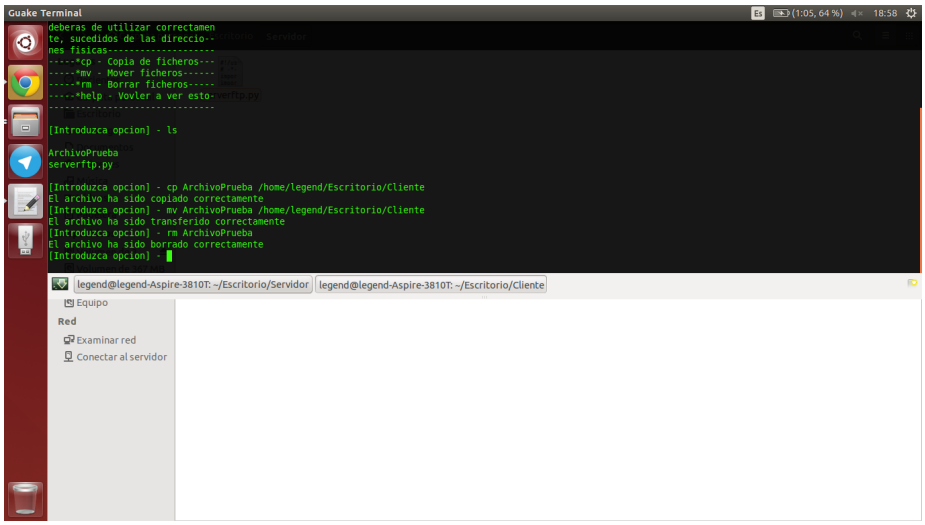

Figura 11: Prueba comando rm

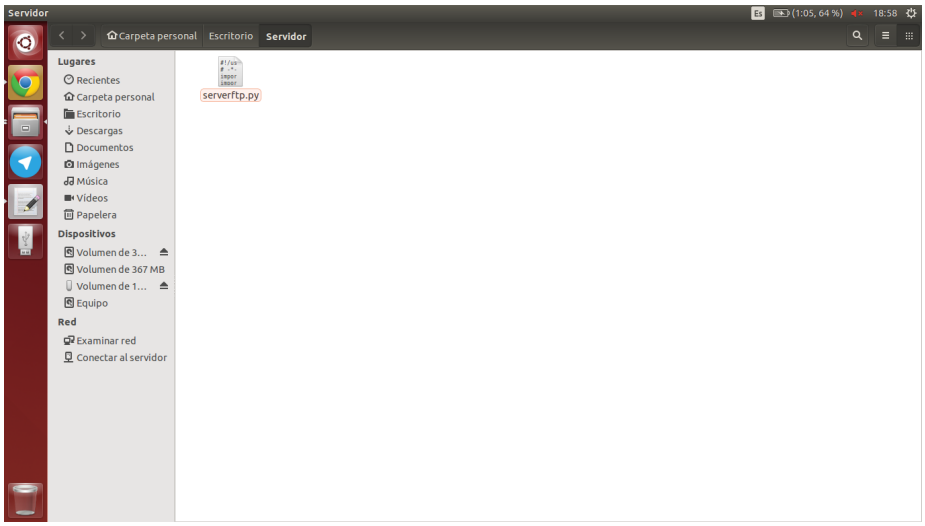

Figura 12: Comprobación del borrado de archivo

Referencias

[1] *Campus de Sistemas Distribuidos.* - https://www.uca.es 2015

[2] *Aprendiendo ZeroMQ.* - http://learning-0mq-with-pyzmq.readthedocs.org/en/latest/ 2015