

Detalles y Documentación sobre el Servidor Y sus varias características

-Patricio Vejar

Modo de uso:

- **Servidor**

Para poder ejecutar el servidor desde su máquina, tiene que tener la estructura de directorios del tipo:

```
├── entrada
├── logs
│   ├── 2025-04-27.txt
│   ├── 2025-04-28.txt
│   └── test2.txt
├── procesados
│   ├── descargaTest.txt
│   ├── documento1 - copia.txt:Zone.Ide
│   ├── documento1.txt
│   ├── documento2.txt
│   ├── enviarTest.txt
│   ├── envio2.txt
│   ├── test1.txt
│   └── test3.txt
├── pyFiles
│   ├── Client.py
│   ├── Server.py
│   ├── pycache
│   │   ├── Server.cpython-312.pyc
│   │   └── demonio.cpython-312.pyc
│   ├── demonio.py
│   ├── grabarSeguridad.txt
│   └── testFile.txt
└── directorios_18_files
```

Los archivos guardados en un directorio aparte y seguido el directorio de entrada, logs, y procesados

Despues simplemente tendra que ejecutar el script: ***Server.py***

Para cerrar el servidor desde la máquina, ingrese el comando: **/EXIT**

Esto cerrará el servidor y cuando todos los usuarios desocupen los puertos el programa terminará definitivamente

- **USUARIO**

Para poder ejecutar el programa de “Cliente” se tiene que asegurar que la ip usada sea la indicada. Puede modificarlo directamente del script o se le dará un prompt en el programa si la ip ingresada no conecta.

Tras ejecutarlo y no tener problemas con la IP.

Accedes al sistema de chat donde podrá hablar con otros usuarios.

Si usa el comando: **/help**

Se le mostrara todos los comandos disponibles y el cómo utilizarlos.

Puede salir del programa utilizando el comando : **/exit**

puede cambiar de servidor utilizando el cambio de ip: **/change**

Puede leer la ip actual del servidor con el comando: **/ip**

Puede leer el puerto actual con el comando : **/prt**

Puede leer los detalles del servidor donde se encuentra con el comando:

/server

Puede ver la lista de archivos actuales en el servir en directorio ENTRADA

con el comando: **/lista**

Podrá examinar los contenidos de un archivo dado en el directorio de

ENTRADA con el comando: **/leer <nombre_archivo>**

Si usted terminó de examinar un archivo y considera que está listo para procesar. Puede enviarlo al directorio PROCESADOS. Con el comando:

/procesar <nombre_archivo>

Puede usted subir de su máquina, desde la carpeta donde el script de Cliente se encuentra un archivo a su elección, para que se aloje en el directorio de

ENTRADA:

/subir <nombre_archivo>

Usted también puede descargar un archivo del servidor en la carpeta de

ENTRADA utilizando el comando: **/descarga <nombre_archivo>**

Y finalmente

Puede ver la actividad del servidor, tales como los acceso a los archivos o los pases de proceso con el comando de logs. Cada día se crea un log asique

puede ver el log completo de un día dado con:

/logs <y-m-d.txt>

Siendo “y” el año, “m” el mes, “d” el día.

Preguntas Varias:

- **Cómo evito las condiciones de carrera del servidor**

Utilizando sistemas de MUTEX controlamos el flujo en que los hilos se ejecutan, una aplicación clara es en el sistema de LOGS, para evitar que se corrompa el documento en el caso, de que se intente acceder a la vez el archivo.

- **Que ventaja tiene usar threads**

Los THREADS se utilizan para aprovechar las capacidades de la CPU y lograr que los procesos se ejecuten simultáneamente, haciendo que el programa sea más rápido y fluido.

La única desventaja es que se tiene que tener más preocupación con el manejo de procesos y el orden en que se ejecutan o terminan.

- **Método de sincronización**

El método de sincronización, MUTEX, es esencialmente un sistema LOCK, o SEMÁFORO BINARIO.

Los procesos bajo la condición de MUTEX, se ejecutan 1 a la vez, solamente empezando el siguiente cuando el primero terminó.

Se crea una lista de procesos con estado “RED” y el primero estado GREEN, una vez que el primero se termina de ejecutar, se remueve de la lista y el nuevo primero pasa a GREEN y así sucesivamente
Este proceso es efectivo al introducirlo en procesos cortos y eficientes.