**EXPLICACIÓN PROGRAMA SERVIDOR-CLIENTE CON EL PROTOCOLO TCP**

El objetivo de este proyecto ha sido crear una comunicación cliente-servidor mediante una solución basada en TCP, donde hemos añadido concurrencia, es decir, nuestro programa permite al usuario crear todos los clientes que quiera dentro de un servidor y permite que haya varios servidores activos al mismo tiempo con distintos clientes a la vez. La interfaz del programa es bastante simple e interactiva en donde una vez iniciados ambos ficheros aparecen dos ventanas para abrir el servidor en un puerto y crear clientes. Si la conexión cliente-servidor es exitosa, se abre una nueva ventana con el nombre de usuario del cliente y este mismo ya puede interactuar con el servidor. Además, hemos buscado la robustez dentro de la aplicación con el fin de que no se den casos extraordinarios durante la ejecución de la aplicación.

Modo de funcionamiento:

1. Ejecuta el script del servidor y elige un puerto.
2. Ejecuta el script del cliente, al crear el usuario necesitarás poner la IP, en este caso, usamos la IP local del ordenador (127.0.0.1), el puerto del servidor activo y el nombre del usuario.
3. Una vez creado el usuario, se te abrirá una nueva ventana con el nombre de usuario introducido y la posibilidad de enviar mensajes al servidor. Los mensajes saldrán en el servidor con el usuario que los ha enviado.

Las palabras claves son:

FIN-> Finaliza la conexión con el servidor y transcurrido 2 segundos se cierra la ventana del usuario.

HORA-> El servidor devuelve la hora actual.

TIEMPO -> El servidor hace una “request” a una API de una web de tiempo y devuelve el tiempo actual en Valencia (temperatura).

1. Como último, cuando se presiona el botón “cerrar servidor” de la ventana servidor, este se encargará de cerrar todas las conexiones con los clientes, y de cerrar la ventana del servidor terminando la ejecución del programa.

Para lograr una aplicación bastante robusta hemos hecho uso de varios elementos:

* Uso de la estructura [try]-[except] en todas las funciones en donde se ejecutan una serie de instrucciones en el (try) y si hay algún error se salta al apartado (except) que dependiendo del contexto del error se hace una cosa u otra.
* Control de la repetición de usuarios en un mismo servidor que se ha logrado mediante la creación de un diccionario en donde guardamos por pares el nombre de usuario y el puerto en el que está conectado y podemos comprobar dado un puerto si ya existe el usuario.
* Comprobación del dato dado por el usuario al crear el puerto, comprobamos que este sea un número entero y sea un número dentro del rango [0,65535]

Por último, aunque usemos el usuario y no el puerto, desde la terminal se va siguiendo el proceso mediante mensajes para que se pueda saber cada usuario a que puerto están asociados.

TRABAJO HECHO POR ALBERTO ANDRÉS Y OSCAR GANDIA.