

UNIVERSIDAD EAFIT  
INGENIERÍA DE SISTEMAS  
PROTOCOLO DE COMUNICACIÓN CLIENTE-SERVIDOR  
PROGRAMACIÓN EN SOCKETS

**Asignatura:** Telemática

**Profesor:** Juan Carlos Montoya Mendoza

**Semestre:** 2020-2

**Integrantes del equipo:** Sebastián Loaiza Correa y Daniel García García.

**Especificación del servicio:** El servicio a implementar será un protocolo de comunicación entre cliente y servidor que usará la ruta de comunicación y los métodos definidos por la librería “socket” de Python, el *server* se inicializará y estará a la escucha de una petición de conexión que hará el *client*, luego de haber establecido dicha conexión, el *client* enviará una petición al *server* que la recibirá, ejecutará dicha petición y mandará un mensaje de respuesta al *client*, este último ciclo de intercambio de información entre el *server* y el *client* se repetirá sucesivamente hasta que el *client* cierre la conexión, sin embargo el *server* seguirá operando independientemente a la espera de un nuevo llamado a conexión.

**Vocabulario de mensajes:** Los siguientes comandos serán los que el *client* podrá enviar al *server* como petición.

- HELLO: Comando de prueba de intercambio de información entre el *client* y el *server*.
- EXIT: Cierra la conexión del *client* actual con el *server*.
- NEW\_BUCKET: Crea un nuevo bucket donde podrá almacenar información (El programa solicitará el nombre deseado para el nuevo bucket).
- LIST\_BUCKET: Muestra al *client* la lista de buckets creados anteriormente.
- DELETE\_BUCKET: Elimina un bucket del *server* (El programa le solicitará el nombre del bucket que desea eliminar).
- UPLOAD\_FILE: Carga un archivo del *client* al *server* (El programa le solicitará el path donde se encuentra el archivo que desea subir, el nombre de dicho archivo y el bucket donde lo desea cargar)
- LIST\_FILE: Muestra al *client* la lista de archivos cargados en un bucket (El programa le solicitará el nombre del bucket donde se encuentran los archivos a listar).
- DOWNLOAD\_FILE: Descarga un archivo del *server* al *client* (El programa le solicitará el bucket donde se encuentra el archivo y el nombre del archivo a descargar).
- DELETE\_FILE: Elimina un archivo del *server* (El programa le solicitará el bucket donde se encuentra el archivo que desea eliminar y el nombre de dicho archivo).

**Nota:** No colocar espacios en blanco o letras distintas antes o después del comando ya que generará automáticamente un error.

**Regla de procedimiento:**

1. Se ejecuta el programa server.py enviando seguidamente como argumento el path donde estarán los buckets, en caso de que no se envíe, el programa lo solicitará posteriormente, y si dicho path no existe, el programa lo creará.
2. Se ejecuta el programa client.py.
3. Se ingresa el comando que desea enviar.
4. Si el comando es incorrecto se notificará el error al usuario y se le solicitará nuevamente ingresar un comando válido.
5. Si el comando es correcto, el programa le solicitará la información necesaria que enviará luego al server mediante un string sin espacios separado por comas.
6. El server interpretará la información enviada por el client y la ejecutará.
7. Si la información enviada es incorrecta o hay un error de ejecución, el server lo notificará.
8. El server enviará la respuesta del comando enviado al client.
9. El client recibirá la información y la mostrará al usuario.
10. El client espera un nuevo comando para repetir los puntos del 3 al 7.
11. Cuando el comando sea EXIT el client cerrará la conexión con el server.
12. El server queda a la espera de una nueva conexión donde se repetirán los pasos del 2 al 9.