Nº	Entrada	Salida	Salida esperada	Propósito y detalles
		REGISTER		
1	REGISTER	Syntax error. Usage: REGISTER <userName>	Syntax error. Usage: REGISTER <userName>	Probar el comando sin introducir el nombre de usuario.
2	REGISTER user1 user2	Syntax error. Usage: REGISTER <userName>	Syntax error. Usage: REGISTER <userName>	Probar el comando introduciendo más de dos argumentos.
3	REGISTER aaa	REGISTER FAIL	REGISTER FAIL	Probar el comando introduciendo un nombre de usuario con 256 caracteres. Ha de fallar ya que el tamaño máximo de nombre de usuario e 256, incluyendo el carácter '\0'
4	REGISTER aaa	REGISTER OK	REGISTER OK	Probar el comando introduciendo un nombre de usuario con 255 caracteres. Un carácter por debajo del límite.
5	REGISTER aaa			Probar el comando introduciendo un nombre de usuario con 257. caracteres. Un carácter por encima del límite.
6	REGISTER test_user1	REGISTER OK	REGISTER OK	Probar el funcionamiento correcto de la función, para más adelante probar la conexión.
7	REGISTER test_user2	REGISTER OK	REGISTER OK	Probar el funcionamiento correcto de la función y crear otro usuario para más adelante probar la conexión.
8	REGISTER test_user3	REGISTER OK	REGISTER OK	Probar el funcionamiento correcto de la función. Este usuario se borrará más adelante.
9	REGISTER test_user1	USERNAME IN USE	USERNAME IN USE	Probar que no registre al usuario si ya existe.
10	REGISTER test_user1	REGISTER FAIL	REGISTER FAIL	Probar a intentar registrar el usuario sin tener el servidor encendido. Encenderlo de nuevo tras el test.
11	REGISTER test_user1	REGISTER FAIL	REGISTER FAIL	Probar a intentar registrar el usuario sin tener el servidor RPC encendido. Encenderlo de nuevo tras el test.
12	REGISTER test_user1	REGISTER FAIL	REGISTER FAIL	Probar a intentar registrar el usuario sin tener el servicio web encendido. Encenderlo de nuevo tras el test.
		UNREGISTER		
13	UNREGISTER	Syntax error. Usage: UNREGISTER <userName>	Syntax error. Usage: UNREGISTER <userName>	Probar el comando sin introducir el nombre de usuario.
14	UNREGISTER user1 user2	Syntax error. Usage: UNREGISTER <userName>	Syntax error. Usage: UNREGISTER <userName>	Probar el comando introduciendo más de dos argumentos.
15	UNREGISTER aa	UNREGISTER FAIL	UNREGISTER FAIL	Probar el comando introduciendo un nombre de usuario con 256 caracteres.
16	UNREGISTER aa	UNREGISTER OK	UNREGISTER OK	Probar el comando introduciendo un nombre de usuario con 255 caracteres.
17	UNREGISTER aaa	UNREGISTER FAIL	UNREGISTER FAIL	Probar el comando introduciendo un nombre de usuario con 257 caracteres.
18	UNREGISTER test_user3	UNREGISTER OK	UNREGISTER OK	Probar el funcionamiento correcto de la función.
19	UNREGISTER test_user3	USER DOES NOT EXIST	USER DOES NOT EXIST	Probar que no desregistre al usuario si no existe.
20	UNREGISTER test_user3	UNREGISTER FAIL	UNREGISTER FAIL	Probar a intentar desregistrar el usuario sin tener el servidor encendido. Encenderlo de nuevo tras el test.
21	UNREGISTER test_user3	UNREGISTER FAIL	UNREGISTER FAIL	Probar a intentar desregistrar el usuario sin tener el servidor RPC encendido. Encenderlo de nuevo tras el test.
22	UNREGISTER test_user3	UNREGISTER FAIL	UNREGISTER FAIL	Probar a intentar desregistrar el usuario sin tener el servicio web encendido. Encenderlo de nuevo tras el test.
		CONNECT		
23	CONNECT	Syntax error. Usage: CONNECT <userName>	Syntax error. Usage: CONNECT <userName>	Probar el comando sin introducir el nombre de usuario.
24	CONNECT user1 user2	Syntax error. Usage: CONNECT <userName>	Syntax error. Usage: CONNECT <userName>	Probar el comando introduciendo más de dos argumentos.

25	CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	CONNECT FAIL	CONNECT FAIL	Probar el comando introduciendo un nombre de usuario con 256 caracteres.	
26	REGISTER aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa DISCONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	REGISTER OK CONNECT OK DISCONNECT OK	REGISTER OK CONNECT OK DISCONNECT OK	Probar el comando introduciendo un nombre de usuario con 255 caracteres. Antes de eso hay que volver a registrarlo. Tras ello, probamos a desconectar el usuario.	
27	CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	CONNECT FAIL	CONNECT FAIL	Probar el comando introduciendo un nombre de usuario con 257 caracteres.	
28	CONNECT test_user3	CONNECT FAIL, USER DOES NOT EXIST	CONNECT FAIL, USER DOES NOT EXIST	Probar a conectar un usuario no registrado.	
29	CONNECT test_user1	CONNECT OK	CONNECT OK	Probar el correcto funcionamiento de la función.	
30	CONNECT test_user2	CONNECT OK	CONNECT OK	Probar el correcto funcionamiento de la función en otra terminal y tener dos usuarios conectados.	En terminal 2
31	REGISTER test_user3 CONNECT test_user3 DISCONNECT test_user3	REGISTER OK CONNECT OK DISCONNECT OK	REGISTER OK CONNECT OK DISCONNECT OK	Probar a conectar un usuario diferente cuando ya hay otro conectado en esa terminal. En este caso se desconecta el usuario que ya hay conectado y se conecta el nuevo. Tras ello, desconectar al usuario que hay conectado para correr los siguientes tests.	
32	CONNECT test_user2	USER ALREADY CONNECTED	USER ALREADY CONNECTED	Probar a conectar en la terminal 1 un usuario que ya está conectado en la terminal 2.	
33	CONNECT test_user2	USER ALREADY CONNECTED	USER ALREADY CONNECTED	Probar a conectar de nuevo el mismo usuario que ya hay conectado en esa terminal.	En terminal 2
34	CONNECT test_user1	CONNECT FAIL	CONNECT FAIL	Probar a intentar conectar el usuario sin tener el servidor encendido. Encenderlo de nuevo tras el test.	
35	CONNECT test_user1	CONNECT FAIL	CONNECT FAIL	Probar a intentar conectar el usuario sin tener el servidor RPC encendido. Encenderlo de nuevo tras el test.	
36	CONNECT test_user1	CONNECT FAIL	CONNECT FAIL	Probar a intentar conectar el usuario sin tener el servicio web encendido. Encenderlo de nuevo tras el test.	
37	CONNECT test_user1 <cerrar y volver a abrir el servidor> CONNECT test_user1	CONNECT OK CONNECT OK	CONNECT OK CONNECT OK	Probar a conectar un usuario e intentar conectarlo de nuevo tras cerrar y volver a abrir el servidor. Al hacerlo, se borran todos los usuarios conectados, por lo que debe permitir la conexión.	
PUBLISH					
38	PUBLISH	Syntax error. Usage: PUBLISH <fileName> <description>	Syntax error. Usage: PUBLISH <fileName> <description>	Probar el comando sin introducir el nombre del achivo ni su descripción.	
39	PUBLISH aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt description1	PUBLISH FAIL	PUBLISH FAIL	Probar el comando con el nombre del archivo teniendo 256 caracteres.	

40	PUBLISH aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt description1	PUBLISH OK	PUBLISH OK	Probar el comando con el nombre del archivo teniendo 255 caracteres.	
41	PUBLISH aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt description1	PUBLISH FAIL	PUBLISH FAIL	Probar el comando con el nombre del archivo teniendo 257 caracteres.	
42	PUBLISH file.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	PUBLISH FAIL	PUBLISH FAIL	Probar el comando con la descripción del archivo teniendo 256 caracteres.	
43	PUBLISH file.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	PUBLISH FAIL	PUBLISH FAIL	Probar el comando con la descripción del archivo teniendo 257 caracteres.	
44	PUBLISH file.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	PUBLISH OK	PUBLISH OK	Probar el comando con la descripción del archivo teniendo 255 caracteres.	
45	PUBLISH file1.txt descripción1	PUBLISH OK	PUBLISH OK	Probar el correcto funcionamiento de la función.	
46	PUBLISH file2.txt descripción2	PUBLISH OK	PUBLISH OK	Probar el correcto funcionamiento de la función. Creado desde el cliente test_user1.	
47	CONNECT test_user2 PUBLISH file3.txt descripción3	CONNECT OK PUBLISH OK	CONNECT OK PUBLISH OK	Probar el correcto funcionamiento de la función. Creado desde otra terminal por el cliente test_user2.	En terminal 2
48	PUBLISH file1.txt descripción1	PUBLISH FAIL, USER DOES NOT EXIST	PUBLISH FAIL, USER DOES NOT EXIST	Probar a publicar cuando el usuario no está registrado. Esto no se puede conseguir con un uso normal, ya que si se intenta publicar un archivo sin haberse conectado a ningún usuario, se obtendrá el error PUBLISH FAIL, USER NOT CONNECTED. Sin embargo, se puede modificar el archivo "users.txt" del servidor para borrar el usuario una vez está conectado. Si esto se hace, se debe obtener el error deseado. Tras este test, se debe restaurar el estado anterior de "users.txt".	
49	DISCONNECT test_user1 PUBLISH file1.txt descripción1 CONNECT test_user1	DISCONNECT OK PUBLISH FAIL, USER NOT CONNECTED CONNECT OK	DISCONNECT OK PUBLISH FAIL, USER NOT CONNECTED CONNECT OK	Probar a publicar sin estar el usuario conectado.	
50	PUBLISH file1.txt descripción1	PUBLISH FAIL, CONTENT ALREADY PUBLISHED	PUBLISH FAIL, CONTENT ALREADY PUBLISHED	Probar a publicar un archivo que ya ha sido publicado por el usuario.	
51	PUBLISH file4.txt descripción4	PUBLISH FAIL	PUBLISH FAIL	Probar a publicar un archivo sin tener el servidor encendido. Encenderlo de nuevo tras el test.	
52	CONNECT test_user1 <cerrar y volver a abrir el servidor> PUBLISH file4.txt descripción4	CONNECT OK PUBLISH FAIL, USER NOT CONNECTED	CONNECT OK PUBLISH FAIL, USER NOT CONNECTED	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo e intentar publicar un archivo. Debe dar error ya que se han desconectado todos los usuarios.	
53	CONNECT test_user1 <cerrar y volver a abrir el servidor> CONNECT test_user1 PUBLISH file4.txt descripción4	CONNECT OK CONNECT OK PUBLISH OK	CONNECT OK CONNECT OK PUBLISH OK	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo, conectarse de nuevo e intentar publicar un archivo.	
54	PUBLISH file5.txt descripción5	PUBLISH FAIL	PUBLISH FAIL	Probar a publicar un archivo sin tener el servidor RPC encendido. Encenderlo de nuevo tras el test.	
55	PUBLISH file5.txt descripción5	PUBLISH FAIL	PUBLISH FAIL	Probar a publicar un archivo sin tener el servicio web encendido. Encenderlo de nuevo tras el test.	

56	CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa PUBLISH aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa CONNECT test_user1	CONNECT OK PUBLISH OK CONNECT OK	CONNECT OK PUBLISH OK CONNECT OK	Probar a publicar un archivo con un nombre de 255 caracteres y una descripción de 255 caracteres desde un usuario con un nombre de 255 caracteres (tamaño máximo de todos los campos).
		DELETE		
57	DELETE	Syntax error. Usage: DELETE <fileName>	Syntax error. Usage: DELETE <fileName>	Probar el comando sin introducir el nombre del achivo.
58	DELETE file1.txt file2.txt	Syntax error. Usage: DELETE <fileName>	Syntax error. Usage: DELETE <fileName>	Probar el comando introduciendo más campos de los requeridos.
59	DELETE aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt	DELETE FAIL	DELETE FAIL	Probar el comando con el nombre del archivo teniendo 256 caracteres.
60	DELETE aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt	DELETE FAIL	DELETE FAIL	Probar el comando con el nombre del archivo teniendo 257 caracteres.
61	DELETE aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt	DELETE OK	DELETE OK	Probar el comando con el nombre del archivo teniendo 255 caracteres.
62	DELETE file2.txt	DELETE OK	DELETE OK	Probar el funcionamiento correcto de la función desde el cliente que publicó ese archivo.
63	DELETE file1.txt	DELETE FAIL, USER DOES NOT EXIST	DELETE FAIL, USER DOES NOT EXIST	Probar a borrar un archivo sin estar el usuario registrado. Esto no se puede conseguir con un uso normal, ya que si se intenta borrar un archivo sin haberse conectado a ningún usuario, se obtendrá el error DELETE FAIL, USER NOT CONNECTED. Sin embargo, se puede modificar el archivo "users.txt" del servidor para borrar el usuario una vez está conectado. Si esto se hace, se debe obtener el error deseado. Tras este test, se debe restaurar el estado anterior de "users.txt".
64	DISCONNECT test_user1 DELETE file1.txt CONNECT test_user1	DISCONNECT OK DELETE FAIL, USER NOT CONNECTED CONNECT OK	DISCONNECT OK DELETE FAIL, USER NOT CONNECTED CONNECT OK	Probar a borrar un archivo sin estar el usuario conectado.
65	DELETE file2.txt	DELETE FAIL, CONTENT NOT PUBLISHED	DELETE FAIL, CONTENT NOT PUBLISHED	Probar a borrar un archivo no publicado: file2.txt se borró anteriormente.
66	DELETE file3.txt	DELETE FAIL, CONTENT NOT PUBLISHED	DELETE FAIL, CONTENT NOT PUBLISHED	Probar a borrar un archivo publicado por otro usuario: file3.txt fue publicado por user_test2.
67	DELETE file1.txt	DELETE FAIL	DELETE FAIL	Probar a borrar un archivo con el servidor apagado. Encenderlo de nuevo tras el test.
68	CONNECT test_user1 <cerrar y volver a abrir el servidor> DELETE file1.txt	CONNECT OK DELET FAIL, USER NOT CONNECTED	CONNECT OK DELET FAIL, USER NOT CONNECTED	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo e intentar borrar un archivo. Debe dar error ya que se han desconectado todos los usuarios.
69	CONNECT test_user1 <cerrar y volver a abrir el servidor> CONNECT test_user1 DELETE file1.txt	CONNECT OK CONNECT OK DELETE OK	CONNECT OK CONNECT OK DELETE OK	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo, conectarse de nuevo e intentar borrar un archivo.

70	DELETE file1.txt	DELETE FAIL	DELETE FAIL	Probar a borrar un archivo con el servidor RPC apagado. Encenderlo de nuevo tras el test.	
71	DELETE file1.txt	DELETE FAIL	DELETE FAIL	Probar a borrar un archivo con el servicio web apagado. Encenderlo de nuevo tras el test.	
72	CONNECT aaa DELETE aa.txt aaa CONNECT test_user1	CONNECT OK DELETE OK CONNECT OK	CONNECT OK DELETE OK CONNECT OK	Probar a borrar un archivo con un nombre de 255 caracteres y una descripción de 255 caracteres desde un usuario con un nombre de 255 caracteres (tamaño máximo de todos los campos).	
LIST_USERS					
73	LIST_USERS arg_test	Syntax error. Use: LIST_USERS	Syntax error. Use: LIST_USERS	Probar el comando introduciendo campos no requeridos.	
74	CONNECT test_user2	CONNECT OK	CONNECT OK	Conectar al test_user2 en la terminal 2 para comprobar que se listan varios usuarios.	En terminal 2
75	LIST_USERS	LIST_USERS OK test_user1 127.0.0.1 41409 test_user2 127.0.0.1 35059	LIST_USERS OK test_user1 <ip> <puerto> test_user2 <ip> <puerto>	Probar el correcto funcionamiento de la función.	
76	CONNECT aaa	CONNECT OK	CONNECT OK	Abrir otra terminal y conectar a un usuario con un nombre de 255 caracteres para comprobar que se lista correctamente.	En terminal 3
77	LIST_USERS	LIST_USERS OK test_user1 127.0.0.1 41409 test_user2 127.0.0.1 35059 aaa 127.0.0.1 38983	LIST_USERS OK test_user1 <ip> <puerto> test_user2 <ip> <puerto> aaa <ip> <puerto>	Probar el correcto funcionamiento de la función cuando un usuario con un nombre de 255 caracteres está conectado.	
78	LIST_USERS	LIST_USERS FAIL, USER DOES NOT EXIST	LIST_USERS FAIL, USER DOES NOT EXIST	Probar a listar los usuarios conectados sin estar el usuario registrado. Esto no se puede conseguir con un uso normal, ya que si se intenta borrar un archivo sin haberse conectado a ningún usuario, se obtendrá el error LIST_USERS FAIL, USER NOT CONNECTED. Sin embargo, se puede modificar el archivo "users.txt" del servidor para borrar el usuario una vez está conectado. Si esto se hace, se debe obtener el error deseado. Tras este test, se debe restaurar el estado anterior de "users.txt".	
79	DISCONNECT test_user1 LIST_USERS CONNECT test_user1	DISCONNECT OK LIST_USERS FAIL, USER NOT CONNECTED CONNECT OK	DISCONNECT OK LIST_USERS FAIL, USER NOT CONNECTED CONNECT OK	Probar a listar los usuarios sin estar conectado.	
80	LIST_USERS	LIST_USERS FAIL	LIST_USERS FAIL	Probar a listar los usuarios sin estar el servidor conectado. Encenderlo de nuevo tras el test.	
81	CONNECT test_user1 <cerrar y volver a abrir el servidor> LIST_USERS	CONNECT OK LIST_USERS FAIL, USER NOT CONNECTED	CONNECT OK LIST_USERS FAIL, USER NOT CONNECTED	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo e intentar listar los usuarios. Debe dar error ya que se han desconectado todos los usuarios.	

82	CONNECT test_user1 <cerrar y volver a abrir el servidor> CONNECT test_user1 LIST_USERS	CONNECT OK CONNECT OK LIST_USERS OK test_user1 127.0.0.1 42921	CONNECT OK CONNECT OK LIST_USERS OK test_user1 <ip> <puerto>	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo, conectarse de nuevo e intentar listar los usuarios conectados. Debe hacerlo correctamente, y tan solo listar al propio test_user1, pues todos los usuarios se desconectaron al cerrar el servidor.	
83	LIST_USERS	LIST_USERS FAIL	LIST_USERS FAIL	Probar a listar los usuarios sin estar el servidor RPC conectado. Encenderlo de nuevo tras el test.	
84	LIST_USERS	LIST_USERS FAIL	LIST_USERS FAIL	Probar a listar los usuarios sin estar el servicio web conectado. Encenderlo de nuevo tras el test.	
			LIST_CONTENT		
85	LIST_CONTENT	Syntax error. Usage: LIST_CONTENT <userName>	Syntax error. Usage: LIST_CONTENT <userName>	Probar el comando sin introducir el campos de nombre de usuario.	
86	LIST_CONTENT test_user1	LIST_CONTENT OK file1.txt descripción1 file4.txt descripción4	LIST_CONTENT OK file1.txt descripción1 file4.txt descripción4	Probar el funcionamiento correcto desde la terminal del usuario test_user1 cuando solicita su propio contenido.	
87	LIST_CONTENT test_user2	LIST_CONTENT OK file3.txt descripción3	LIST_CONTENT OK file3.txt descripción3	Probar el funcionamiento correcto desde la terminal del usuario test_user1 cuando solicita el contenido de test_user2.	
88	CONNECT test_user2 LIST_CONTENT test_user1	CONNECT OK LIST_CONTENT OK file1.txt descripción1 file4.txt descripción4	CONNECT OK LIST_CONTENT OK file1.txt descripción1 file4.txt descripción4	Probar el funcionamiento correcto desde la terminal del usuario test_user2 cuando solicita el contenido de test_user1.	En terminal 2
89	CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa PUBLISH aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa CONNECT test_user1	CONNECT OK PUBLISH OK CONNECT OK	CONNECT OK PUBLISH OK CONNECT OK	Probar a publicar un archivo con un nombre de 255 caracteres y una descripción de 255 caracteres desde un usuario con un nombre de 255 caracteres (tamaño máximo de todos los campos) para después listarlo.	
90	LIST_CONTENT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	LIST_CONTENT OK aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaa.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	LIST_CONTENT OK aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaa.txt aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	Probar a listar el contenido de un usuario con 255 caracteres que ha publicado un archivo con un nombre y una descripción de 255 caracteres.	
91	LIST_CONTENT test_user2	LIST_CONTENT FAIL, USER DOES NOT EXIST	LIST_CONTENT FAIL, USER DOES NOT EXIST	Probar a listar el contenido sin que el usuario que realiza la operación esté registrado. Esto no se puede conseguir con un uso normal, ya que si se intenta borrar un archivo sin haberse conectado a ningún usuario, se obtendrá el error LIST_CONTENT FAIL, USER NOT CONNECTED. Sin embargo, se puede modificar el archivo "users.txt" del servidor para borrar el usuario una vez está conectado. Si esto se hace, se debe obtener el error deseado. Tras este test, se debe restaurar el estado anterior de "users.txt".	
92	DISCONNECT test_user1 LIST_CONTENT test_user2 CONNECT test_user1	DISCONNECT OK LIST_CONTENT FAIL, USER NOT CONNECTED CONNECT OK	DISCONNECT OK LIST_CONTENT FAIL, USER NOT CONNECTED CONNECT OK	Probar a listar el contenido sin que el usuario que realiza la operación esté conectado.	
93	LIST_CONTENT test_user3	LIST_CONTENT FAIL, REMOTE USER DOES NOT EXIST	LIST_CONTENT FAIL, REMOTE USER DOES NOT EXIST	Probar a listar el contenido de un usuario que no está registrado en el servicio.	

94	LIST_CONTENT test_user2	LIST_CONTENT FAIL	LIST_CONTENT FAIL	Probar a listar el contenido de un usuario sin que el servidor esté funcionando. Encenderlo de nuevo tras el test.	
95	CONNECT test_user1 <cerrar y volver a abrir el servidor> LIST_CONTENT test_user2	CONNECT OK LIST_CONTENT FAIL, USER NOT CONNECTED	CONNECT OK LIST_CONTENT FAIL, USER NOT CONNECTED	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo e intentar listar el contenido publicado por un usuario. Debe dar error ya que se han desconectado todos los usuarios.	
96	CONNECT test_user1 <cerrar y volver a abrir el servidor> CONNECT test_user1 LIST_CONTENT test_user2	CONNECT OK CONNECT OK LIST_CONTENT OK file3.txt descripción3	CONNECT OK CONNECT OK LIST_CONTENT OK file3.txt descripción3	Probar a conectarse, cerrar el servidor para desconectar a todos los usuarios, volver a abrirlo, conectarse de nuevo e intentar listar el contenido publicado por un usuario.	
97	LIST_CONTENT test_user2	LIST_CONTENT FAIL	LIST_CONTENT FAIL	Probar a listar el contenido de un usuario sin que el servidor RPC esté funcionando. Encenderlo de nuevo tras el test.	
98	LIST_CONTENT test_user2	LIST_CONTENT test_user2	LIST_CONTENT test_user2	Probar a listar el contenido de un usuario sin que el servicio web esté funcionando. Encenderlo de nuevo tras el test.	
GET_FILE					
99	GET_FILE	Syntax error. Usage: GET_FILE <userName> <remote_fileName> <local_fileName>	Syntax error. Usage: GET_FILE <userName> <remote_fileName> <local_fileName>	Probar el comando sin introducir los campos necesarios.	
100	QUIT python3 client.py -s <ip del servidor> -p <puerto del servidor> GET_FILE test_user2 file3.txt file3_local.txt	+++ FINISHED +++ GET_FILE FAIL	+++ FINISHED +++ GET_FILE FAIL	Probar a transferir un archivo desde test_user2 a test_user1 sin que el segundo esté conectado. En lugar de desconectarlo mediante DISCONNECT, se utiliza QUIT porque el siguiente test requiere no haber corrido antes LIST_USERS.	
101	CONNECT test_user1 GET_FILE test_user2 file3.txt file3_local.txt	CONNECT OK GET_FILE FAIL	CONNECT OK GET_FILE FAIL	Probar a transferir un archivo desde test_user2 sin que este esté conectado, y sin haber hecho previamente LIST_USERS.	
102	LIST_USERS GET_FILE test_user2 file3.txt file3_local.txt	LIST_USERS OK test_user1 127.0.0.1 33003 GET_FILE FAIL	LIST_USERS OK test_user1 <ip> <puerto> GET_FILE FAIL	Probar a transferir un archivo desde test_user2 sin que este esté conectado, pero habiendo hecho previamente LIST_USERS.	
103	CONNECT test_user2	CONNECT OK	CONNECT OK	Conectar al test_user2 en la terminal 2 para los posteriores tests.	En terminal 2
104	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE FAIL	GET_FILE FAIL	Probar a transferir un archivo desde test_user2 después de que este se haya conectado, pero sin haber hecho LIST_USERS después de que se conectara.	
105	LIST_USERS GET_FILE test_user2 file_fake.txt file3_local.txt	LIST_USERS OK test_user1 127.0.0.1 33003 test_user2 127.0.0.1 59187 GET_FILE FAIL / FILE NOT EXIST	LIST_USERS OK test_user1 <ip> <puerto> test_user2 <ip> <puerto> GET_FILE FAIL / FILE NOT EXIST	Probar a transferir un archivo desde test_user2 después de que este se haya conectado y habiendo hecho LIST_USERS después de que se conectara. En este caso, se intenta conseguir un archivo que ni existe en el ordenador de test_user2 ni lo tiene listado.	
106	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE FAIL / FILE NOT EXIST	GET_FILE FAIL / FILE NOT EXIST	Probar a transferir un archivo desde test_user2 después de que este se haya conectado y habiendo hecho LIST_USERS después de que se conectara. En este caso, se intenta conseguir un archivo que no existe en el ordenador de test_user2, pero sí lo tiene listado.	
107	GET_FILE test_user2 file4.txt file4_local.txt	GET_FILE FAIL / FILE NOT EXIST	GET_FILE FAIL / FILE NOT EXIST	Probar a transferir un archivo desde test_user2 después de que este se haya conectado y habiendo hecho LIST_USERS después de que se conectara. En este caso, se intenta conseguir un archivo que sí existe en el ordenador de test_user2, pero no lo tiene listado. Para ello, se tiene que crear file4.txt en el ordenador de test_user2.	
108	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE OK	GET_FILE OK	Probar a transferir un archivo desde test_user2 después de que este se haya conectado y habiendo hecho LIST_USERS después de que se conectara. En este caso, se intenta conseguir un archivo que sí existe en el ordenador de test_user2 y también lo tiene listado. Para ello, se tiene que crear file3.txt en el ordenador de test_user2.	
109	CONNECT aa	CONNECT OK	CONNECT OK	Conectar al usuario con un nombre de 255 caracteres en la terminal 3 para comprobar que sus archivos se transfieren correctamente.	En terminal 3

110	LIST_USERS GET_FILE aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt local_aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt	LIST_USERS OK test_user1 127.0.0.1 33003 test_user2 127.0.0.1 59187 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 127.0.0.1 42481 GET_FILE OK	LIST_USERS OK test_user1 <ip> <puerto> test_user2 <ip> <puerto> aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa <ip> <puerto> GET_FILE OK	Probar a transferir un archivo con un nombre de 255 caracteres desde el usuario de 255 caracteres y guardarlo localmente con un nombre de archivo de 255 caracteres. Se ha de crear el archivo en el ordenador del usuario remoto.	
111	GET_FILE aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt local_aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.txt	GET_FILE FAIL	GET_FILE FAIL	Probar a transferir un archivo con un nombre de 255 caracteres desde el usuario de 255 caracteres y guardarlo localmente con un nombre de archivo de 256 caracteres.	
112	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE OK	GET_FILE OK	Probar a pedir un archivo con el servidor apagado. Encenderlo de nuevo tras el test. Ha de funcionar, ya que la transferencia es peer-to-peer y el servidor no se ve involucrado.	
113	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE OK	GET_FILE OK	Probar a pedir un archivo con el servidor RPC apagado. Encenderlo de nuevo tras el test. Ha de funcionar, ya que el servidor RPC no se ve involucrado.	
114	GET_FILE test_user2 file3.txt file3_local.txt	GET_FILE OK	GET_FILE OK	Probar a pedir un archivo con el servicio web apagado. Encenderlo de nuevo tras el test. Ha de funcionar, ya que el servicio web no se ve involucrado.	
DISCONNECT					
115	DISCONNECT	Syntax error. Usage: DISCONNECT <userName>	Syntax error. Usage: DISCONNECT <userName>	Probar el comando sin introducir el usuario a desconectar.	
116	DISCONNECT user1 user2	Syntax error. Usage: DISCONNECT <userName>	Syntax error. Usage: DISCONNECT <userName>	Probar el comando introduciendo varios usuarios a la vez.	
117	CONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa DISCONNECT aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	CONNECT OK DISCONNECT OK	CONNECT OK DISCONNECT OK	Probar a conectar y desconectar un usuario con 255 caracteres.	
118	DISCONNECT test_user1	DISCONNECT FAIL / USER NOT CONNECTED	DISCONNECT FAIL / USER NOT CONNECTED	Probar a desconectar test_user1 sin que haya nadie conectado.	
119	CONNECT test_user2	CONNECT OK	CONNECT OK	Conectar test_user2 en la terminal 2.	En terminal 2
120	DISCONNECT test_user2	DISCONNECT FAIL / USER NOT CONNECTED	DISCONNECT FAIL / USER NOT CONNECTED	Probar a desconectar test_user2 sin que haya nadie conectado en esta terminal.	
121	CONNECT test_user1	CONNECT OK	CONNECT OK	Conectar al test_user1 para realizar los posteriores tests.	
122	DISCONNECT test_user2	DISCONNECT FAIL / USER NOT CONNECTED	DISCONNECT FAIL / USER NOT CONNECTED	Probar a desconectar test_user2 estando test_user1 conectado en esta terminal.	
123	DISCONNECT test_user1	DISCONNECT FAIL / USER DOES NOT EXIST	DISCONNECT FAIL / USER DOES NOT EXIST	Probar a desconectar un usuario que no esté registrado. Esto no se puede conseguir con un uso normal, ya que si se intentadesconectar un usuario sin haberse conectado a ninguno, se obtendrá el error DISCONNECT FAIL / USER NOT CONNECTED. Sin embargo, se puede modificar el archivo "users.txt" del servidor para borrar el usuario una vez está conectado. Si esto se hace, se debe obtener el error deseado. Tras este test, se debe restaurar el estado anterior de "users.txt".	
124	DISCONNECT test_user1	DISCONNECT OK	DISCONNECT OK	Probar el funcionamiento correcto de la función.	

125	CONNECT test_user1 <cerrar servidor> DISCONNECT test_user1	CONNECT OK DISCONNECT OK	CONNECT OK DISCONNECT OK	Probar a desconectar test_user1 con el servidor cerrado. Se ha de permitir la operación, ya que al estar cerrado el servidor, de cara e este el usuario ya está desconectado, pues al reiniciarlo se borra el registro de usuarios conectados. De este modo, se le permite al usuario desconectarse para dejar de transmitir. Encenderlo el servidor de nuevo tras el test.
126	CONNECT test_user1 <cerrar servidor RPC> DISCONNECT test_user1	CONNECT OK DISCONNECT FAIL	CONNECT OK DISCONNECT FAIL	Probar a desconectar un usuario con el servidor RPC apagado. Encenderlo de nuevo tras el test.
127	DISCONNECT test_user1	DISCONNECT FAIL	DISCONNECT FAIL	Probar a desconectar un usuario con el servicio web apagado. Encenderlo de nuevo tras el test.
			QUIT / <CTRL+C>	
128	QUIT	+++ FINISHED +++	+++ FINISHED +++	Probar a salir del programa cuando mientras test_user1 está conectado.
129	python 3 client.py -s <ip del servidor> -p <puerto del servidor> QUIT	 +++ FINISHED +++	 +++ FINISHED +++	Probar a salir del programa mientras no haya un usuario conectado.
130	python 3 client.py -s <ip del servidor> -p <puerto del servidor> <cerrar el servidor> QUIT	 +++ FINISHED +++	 +++ FINISHED +++	Probar a salir del programa mientras no hay un usuario conectado y el servidor está apagado. Encenderlo de nuevo tras el test.
131	python 3 client.py -s <ip del servidor> -p <puerto del servidor> <cerrar el servidor RPC> QUIT	 +++ FINISHED +++	 +++ FINISHED +++	Probar a salir del programa mientras no hay un usuario conectado y el servidor RPC está apagado. Encenderlo de nuevo tras el test.
132	python 3 client.py -s <ip del servidor> -p <puerto del servidor> <cerrar el servicio web> QUIT	 +++ FINISHED +++	 +++ FINISHED +++	Probar a salir del programa mientras no hay un usuario conectado y el servicio web está apagado. Encenderlo de nuevo tras el test.
133	python 3 client.py -s <ip del servidor> -p <puerto del servidor> CONNECT test_user1 <cerrar el servidor> QUIT	 CONNECT OK +++ FINISHED +++	 CONNECT OK +++ FINISHED +++	Probar a salir del programa mientras test_user1 está conectado y el servidor está apagado. Encenderlo de nuevo tras el test.
134	python 3 client.py -s <ip del servidor> -p <puerto del servidor> CONNECT test_user1 <cerrar el servidor RPC> QUIT	 CONNECT OK +++ FINISHED +++	 CONNECT OK +++ FINISHED +++	Probar a salir del programa mientras test_user1 está conectado y el servidor RPC está apagado. Encenderlo de nuevo tras el test. Se permite la desconexión en el lado del cliente para que todos los hilos se cierren y se pueda parar la ejecución. Sin embargo, en el servidor el registro de usuarios conectados no se actualiza, por lo que se genera una inconsistencia: el usuario aparece conectado para el servidor aunque ya no lo está. Esta inconsistencia hará que el usuario no pueda volver a conectarse hasta que se reinicie el servidor o se elimine a mano el usuario de "users.txt". No hay forma de solucionar esto siguiendo el protocolo requerido.
135	python 3 client.py -s <ip del servidor> -p <puerto del servidor> CONNECT test_user1 <cerrar el servicio web> QUIT	 CONNECT OK +++ FINISHED +++	 CONNECT OK +++ FINISHED +++	Probar a salir del programa mientras test_user1 está conectado y el servicio web está apagado. Se permitirá la desconexión y esta se verá reflejada en el registro del servidor. En el servidor RPC se imprimirá "test_user1 DISCONNECT 01/01/1970 00:00:00". Se envía el comienzo del registro del tiempo UNIX debido a que el servicio web está caído, pero se quiere forzar la desconexión.