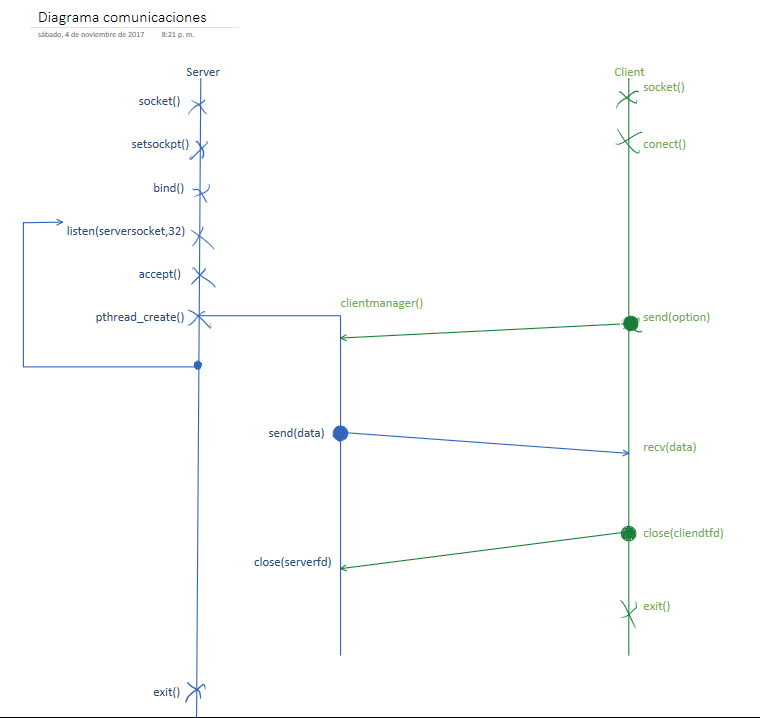
Kevin Alejandro Vanegas

1018483781

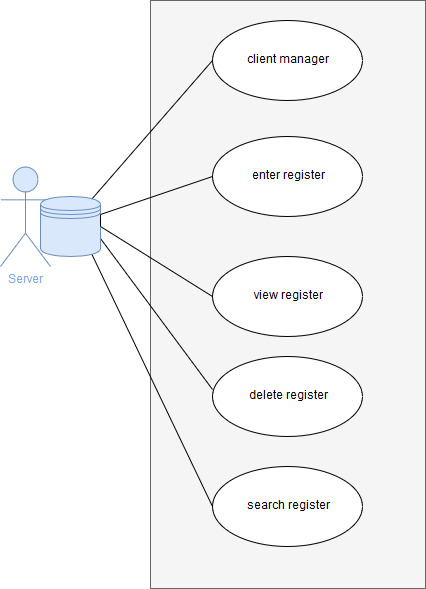
Noviembre, 2017



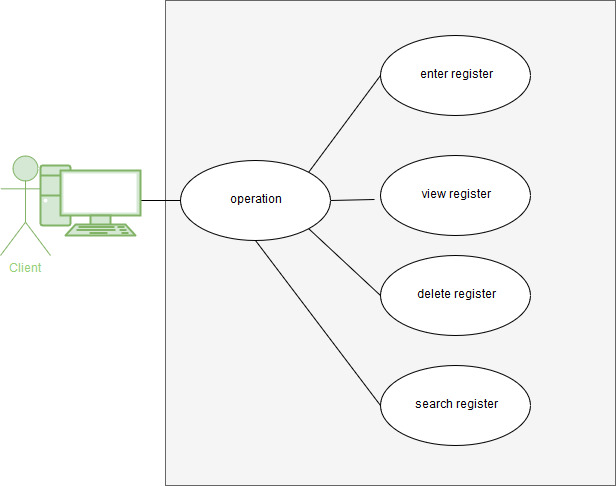
Comunication diagram



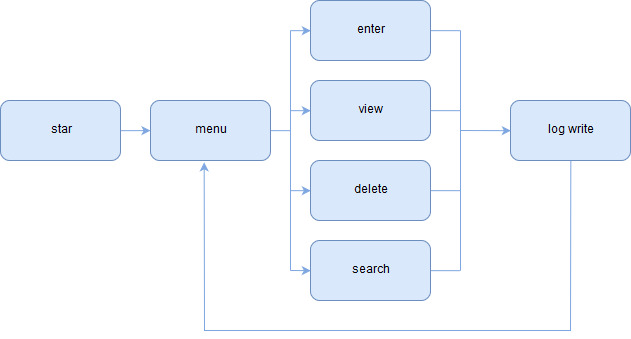
Use case diagram server



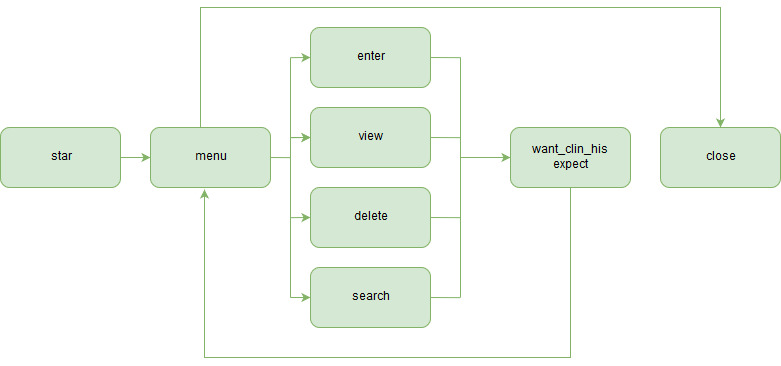
Use case diagram client



Block diagram server



Block diagram client



Mediante los programas p2-dogClient.c y p2-dogServer.c, se gestiona los registros de mascotas de una veterinaria. Estos registros son almacenados en el disco, el servidor recibe peticiones de los clientes a través de la red para realizar solicitudes de insertar, leer, borrar y buscar. Al entrar en ejecución el servidor no publica ningún mensaje, este solamente se limita a gestionar las operaciones con los clientes y estas operaciones son almacenadas en serverDogs.log que han ocurrido. El número de clientes que se pueden conectar al servidor son 32, predefinido el puerto 3566, al estar editando, el hilo bloquea los archivos que resultaran modificados para así garantizar que otro no los modifique al tiempo que lo hace el primero.

En el cliente se muestra un menú donde:

1. **Ingresar registro**. Al ingresar pide uno a uno, los campos de un registro.

Se crea la estructura dogType asignándole espacio en memoria, luego que el usuario digita los datos pedidos se reciben esos datos al final del archivo dataDogs.dat y se envían a través de soket (send), cuando es recibido satisfactoriamente se libera la memoria usada y pide una confirmación digitando cualquier tecla.

2. **Ver registro.** Al ingresar muestra el número de registros presentes y

solicita el número del registro a ver. Valida que el número sea válido.

Se cuentan el número de registros existentes en el archivo dataDogs.dat, enviándose al cliente dicho número, el cliente publica dicho número para luego recibir un número de registro determinado por el usuario, este número es enviado al servidor, donde devuelve los datos de la mascota que el usuario deseo ver. Se crea un archivo tipo log donde se encuentra la historia clínica de la mascota recibida para finalmente ser abierta en gedit en la parte del cliente. Al ser editada la historia clínica se envía de nuevo al servidor donde se guarda y envía al cliente una confirmación donde elimina el archivo tipo log y publica una confirmación.

3. **Borrar registro**. Al ingresar muestra el número de registros presentes y

solicita el número del registro a borrar. El registro es borrado del archivo,

por lo que el archivo debe reducir su tamaño.

Se cuentan la cantidad de registros existentes en dataDogs.dat y este dato es enviado al cliente donde lo recibe y publica. Pide un número de registro a eliminar y cuando este es digitado es enviado de vuelta. En el sistema toda la información incluida la historia clínica se borra, al hacer esto es enviado una confirmación al cliente donde publica que fue exitoso.

4. **Buscar registro.** Solicita una cadena de caracteres a buscar en los campos

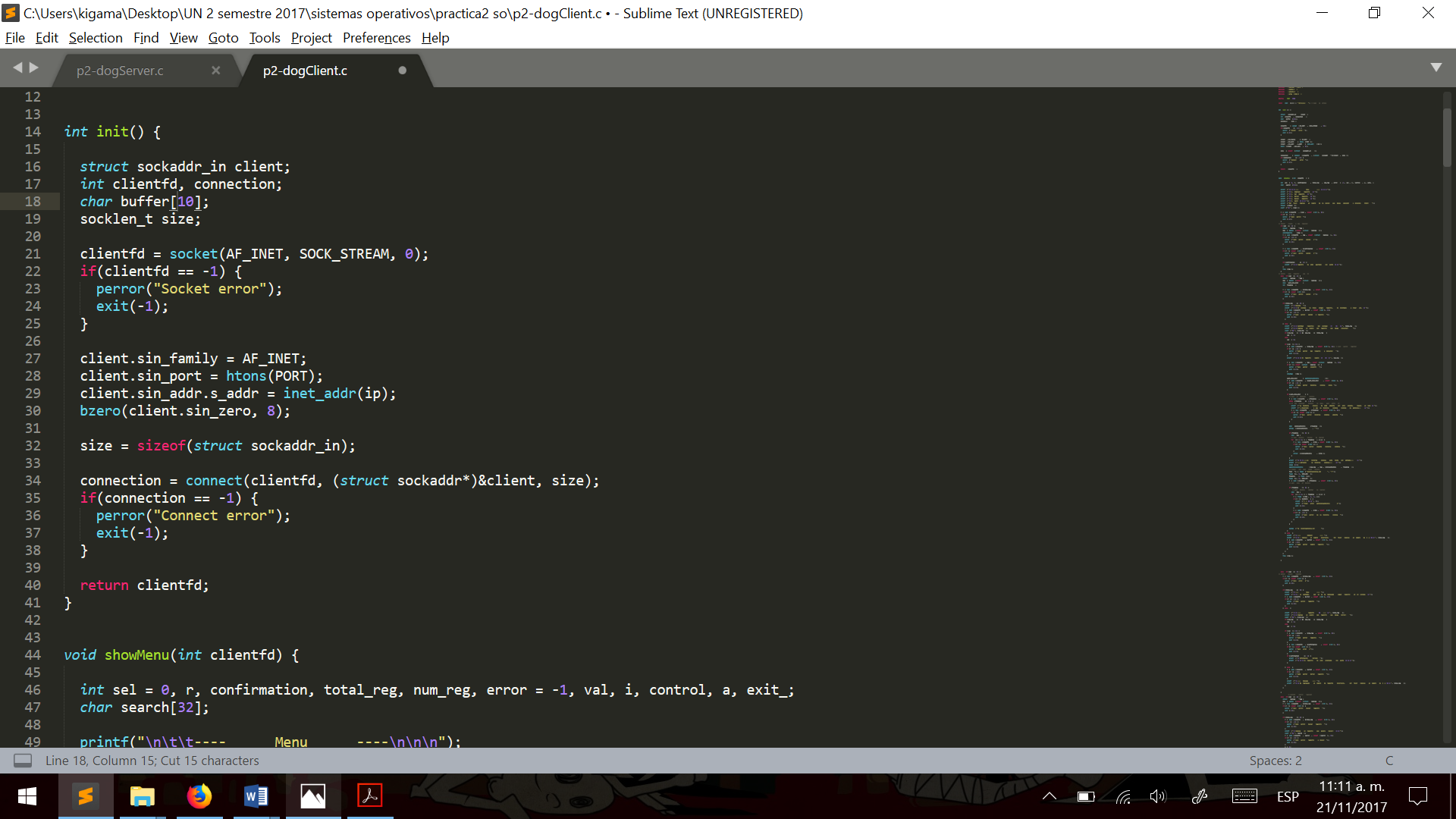
nombre de los registros. Muestra todos los registros que coincidan

completamente con el nombre. No se distingue mayúsculas de minúsculas.

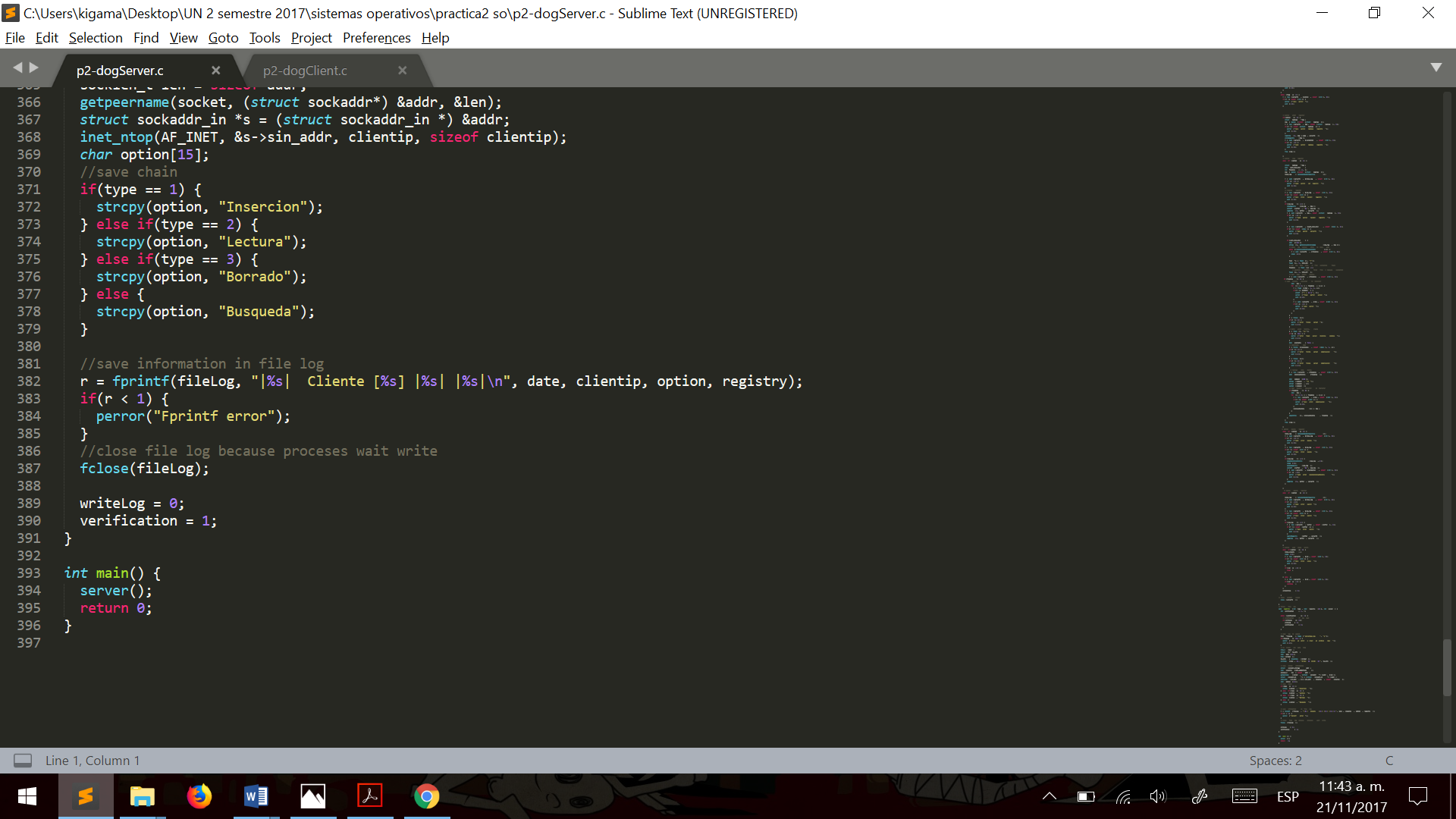
Se cuentan la cantidad de registros existentes en dataDogs.dat y este dato es enviado al cliente donde lo recibe y publica. Recibe el nombre de la mascota recordando que es indistinguible para el sistema las mayúsculas y minúsculas y es enviado de vuelta, revisa en dataDogs.dat las coincidencias que existen, en caso de existir regresa la información de este al cliente donde lo publica.

5. **Salir.** Se cierra la conexión existente del cliente con el servidor y se cierra también el soket donde se da un mensaje donde se termina la conexión.

Cuando una opción es digitada en el cliente, esta es enviada al servidor a fin de iniciar su gestión, la transferencia de procesos se hará mediante la escritura directa de la estructura. La transferencia se hace carácter por carácter.



Ya definidos el puerto y la ip, utilizando connect para hacer la llamada al sistema por un descriptor de archivo a dirección especificada

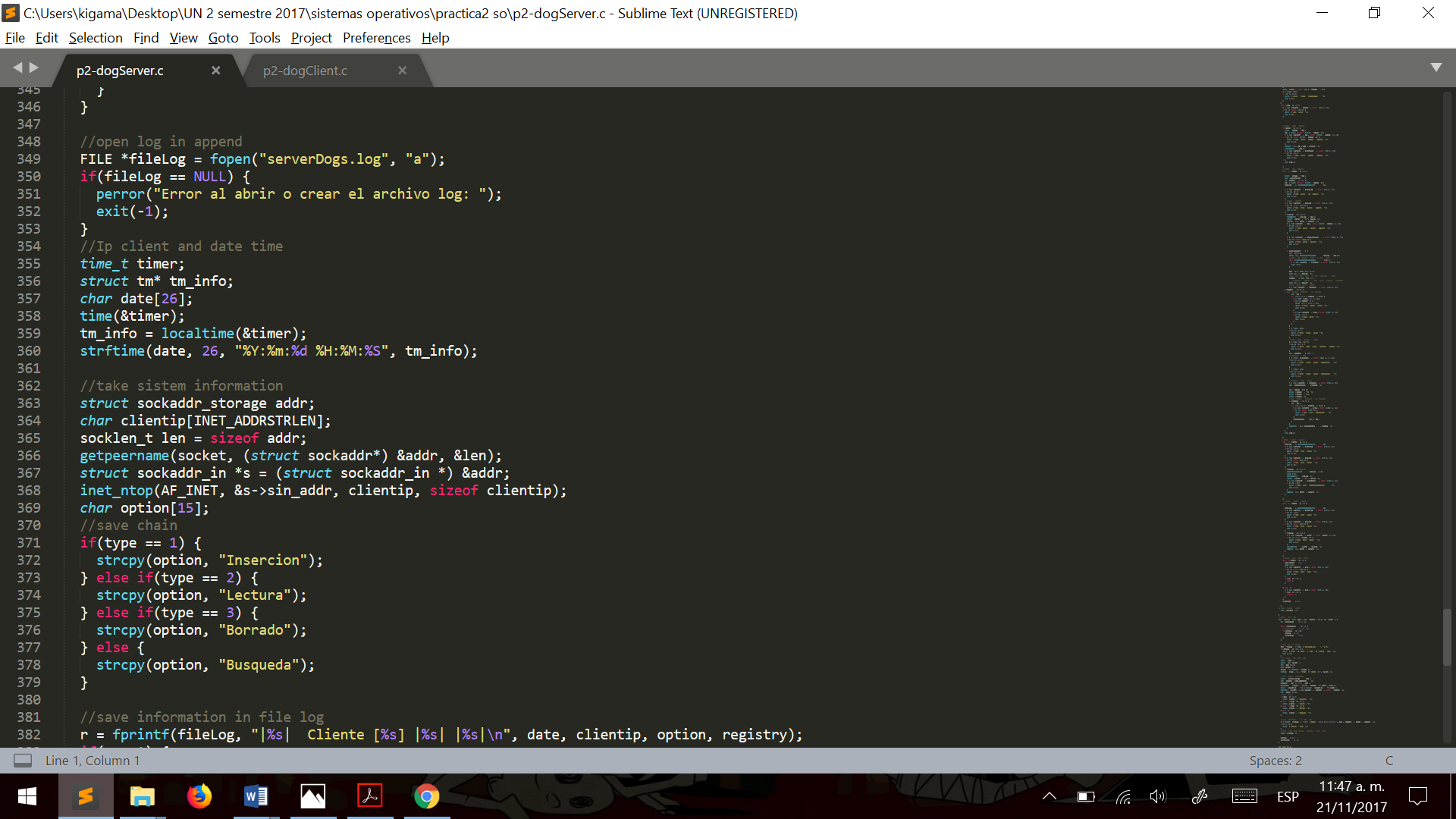


En esta línea se define el formato del archivo log, donde se llena como lo exigido.

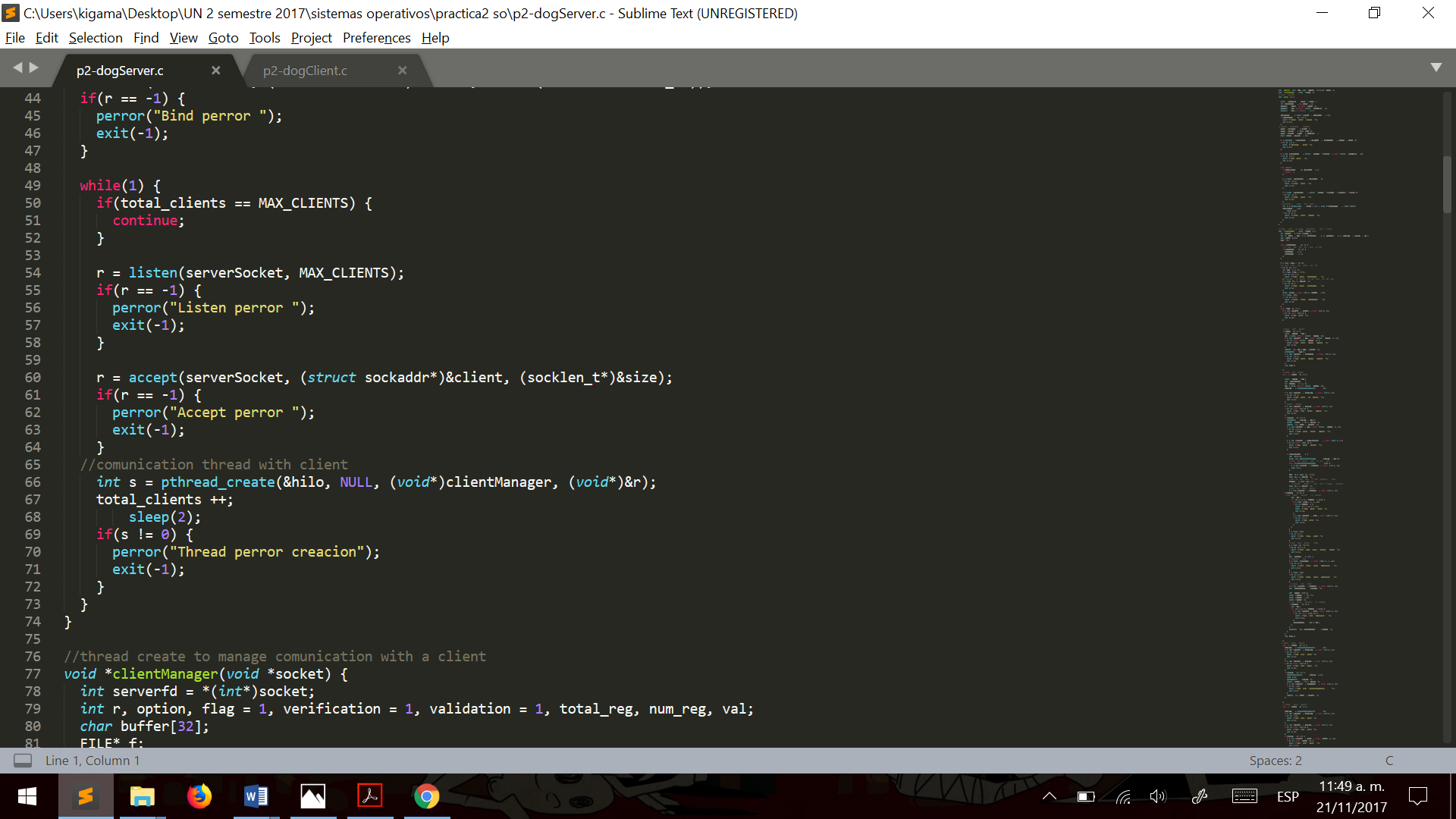
Logwrite es una función global donde se hace el trato de este archivo log

[Fecha YYYYMMDDTHHMMSS] Cliente [IP] [inserción | lectura | borrado | búsqueda] [registro

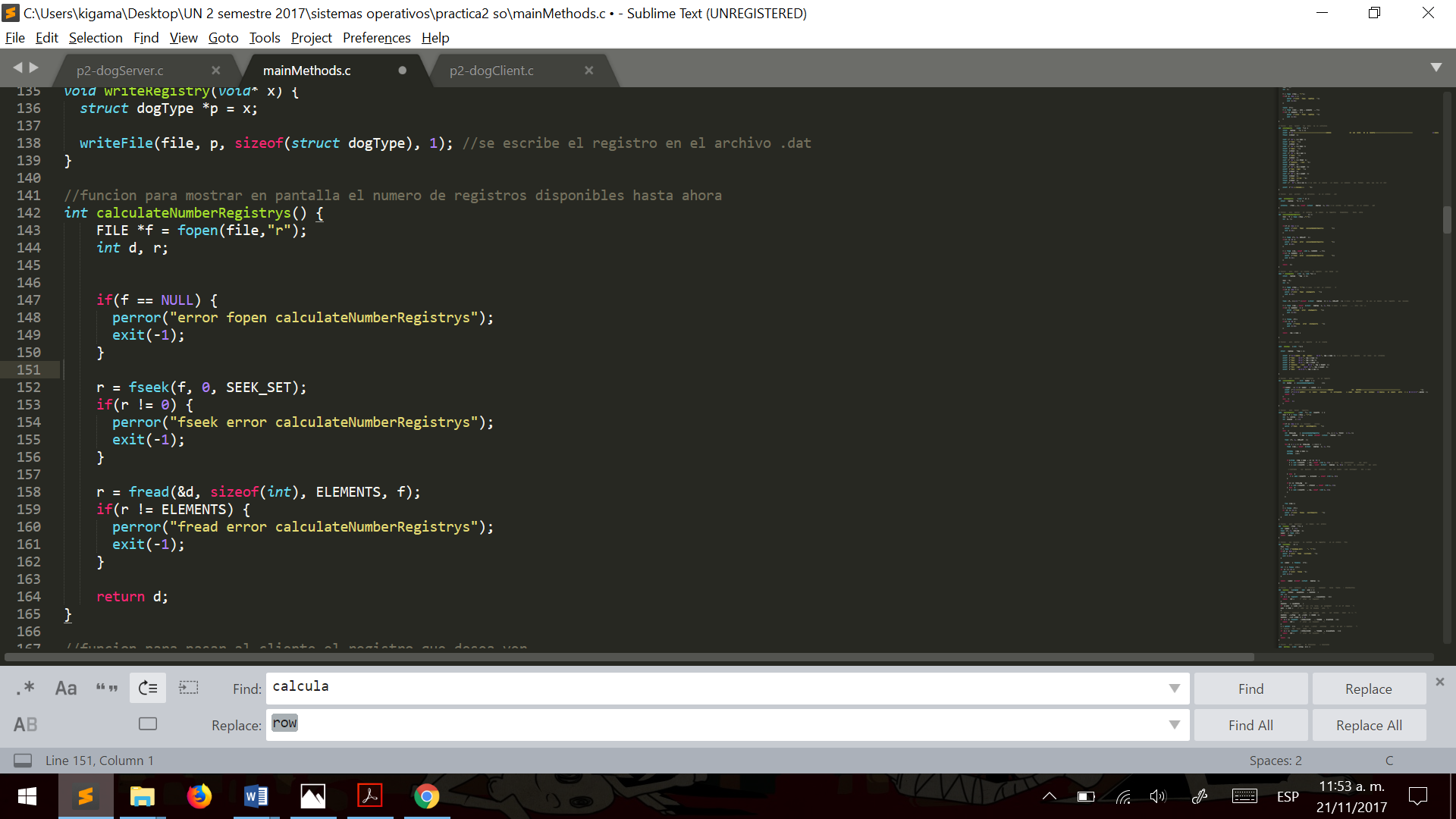
| cadena buscada ]



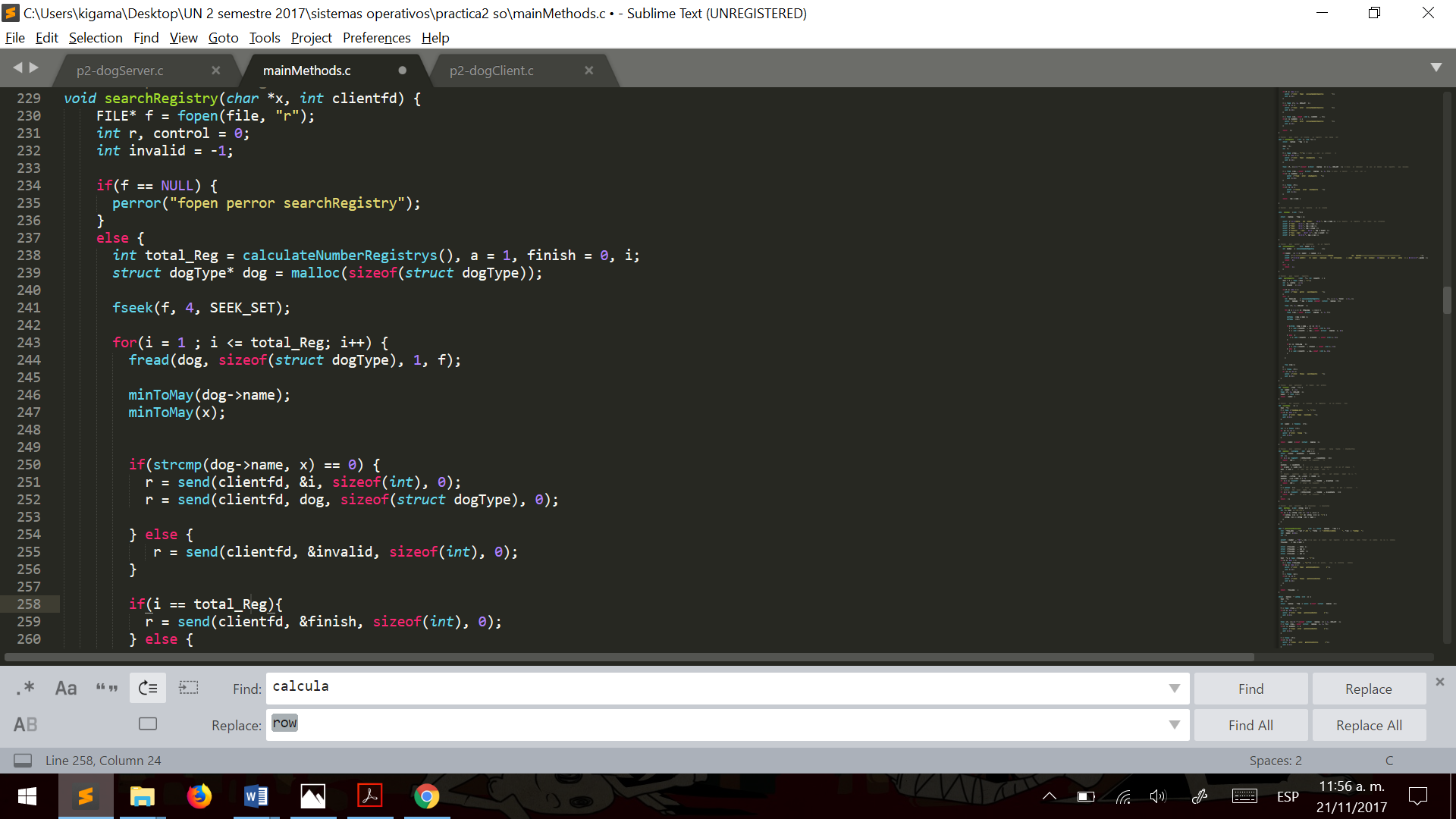
Donde se obtiene la información del sistema.



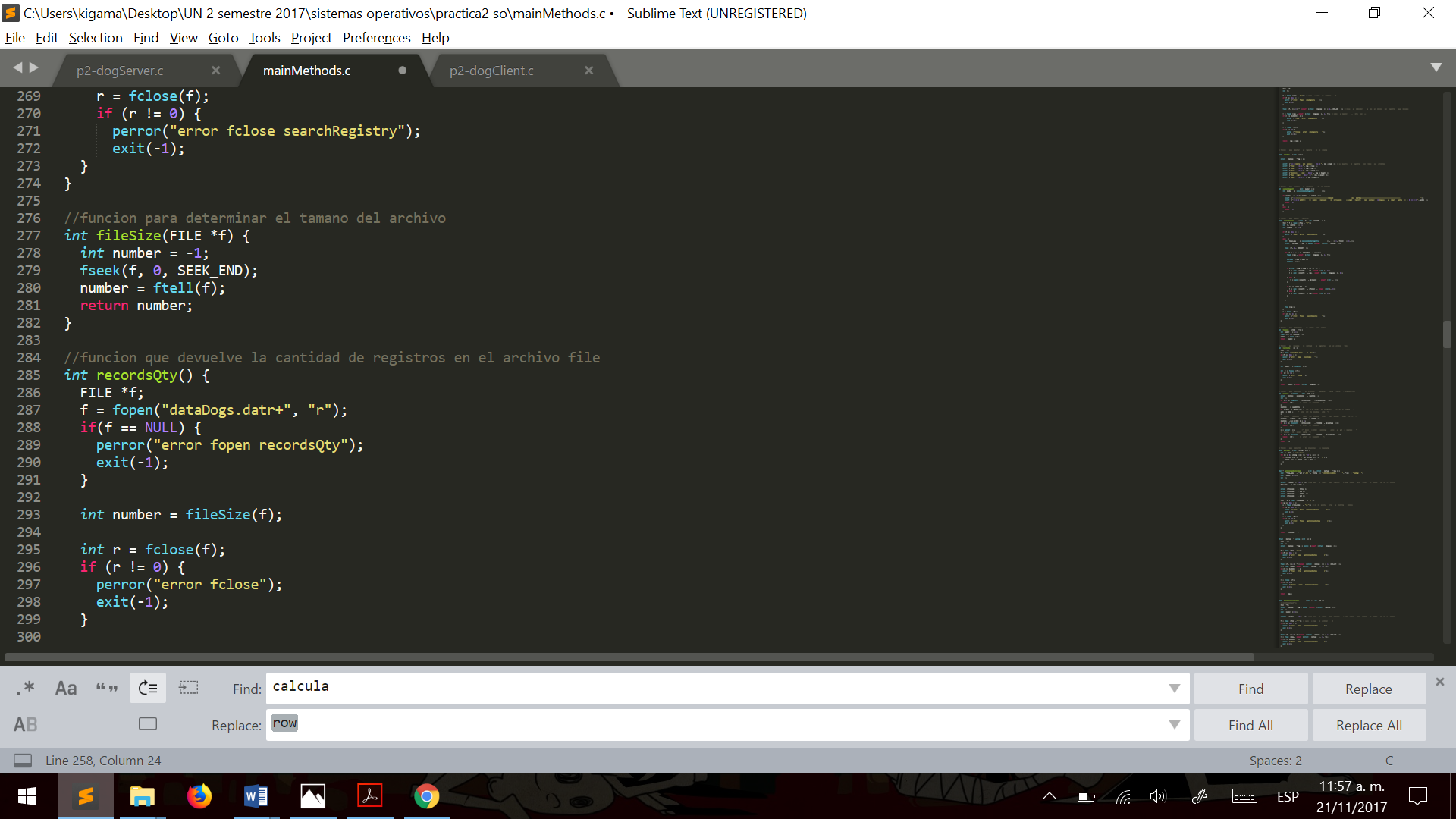
Se hace el trato de número de clientes conectados.



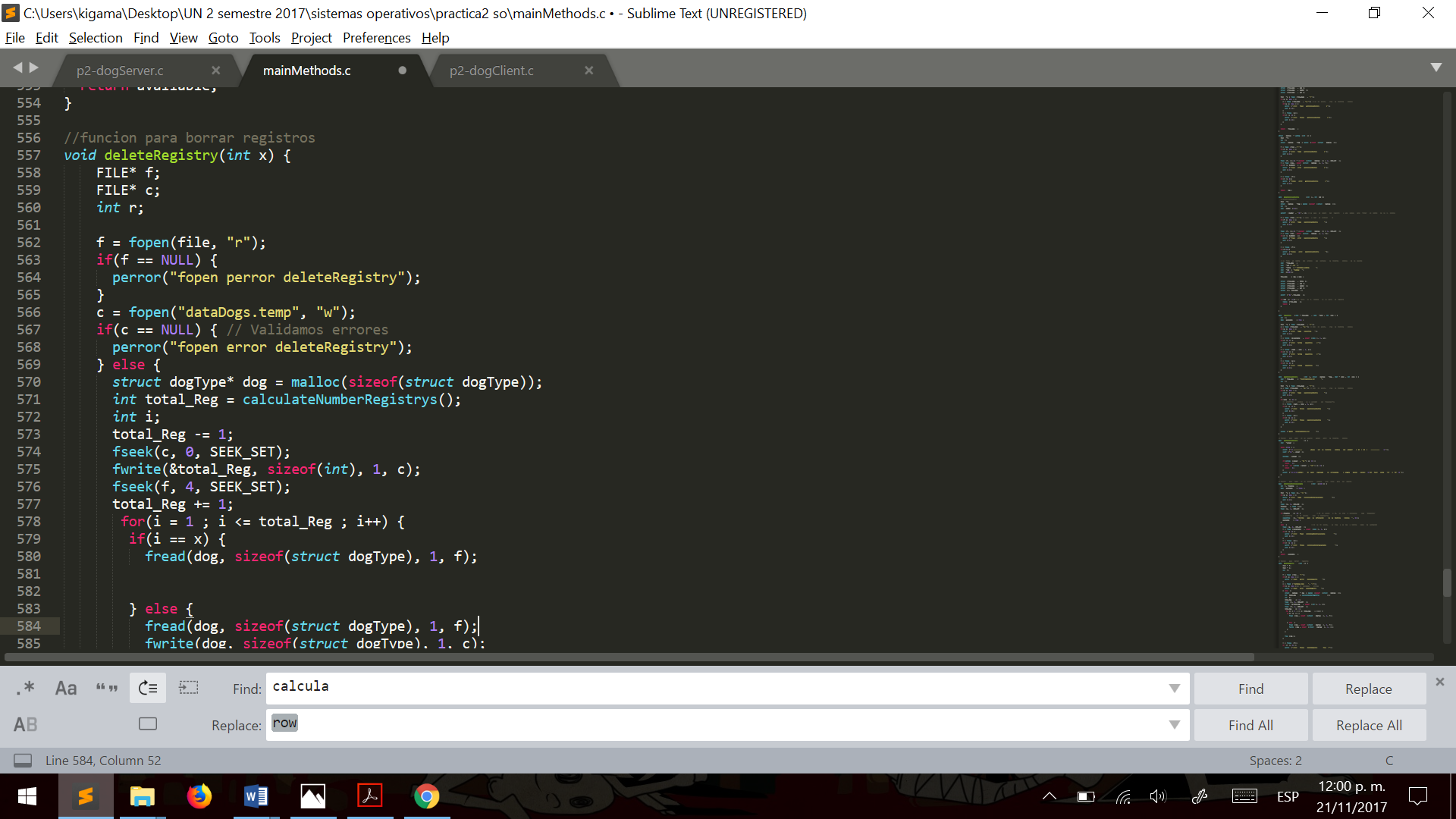
Esta función es casi usada en todas las opciones del menú, cuenta el número de registros.



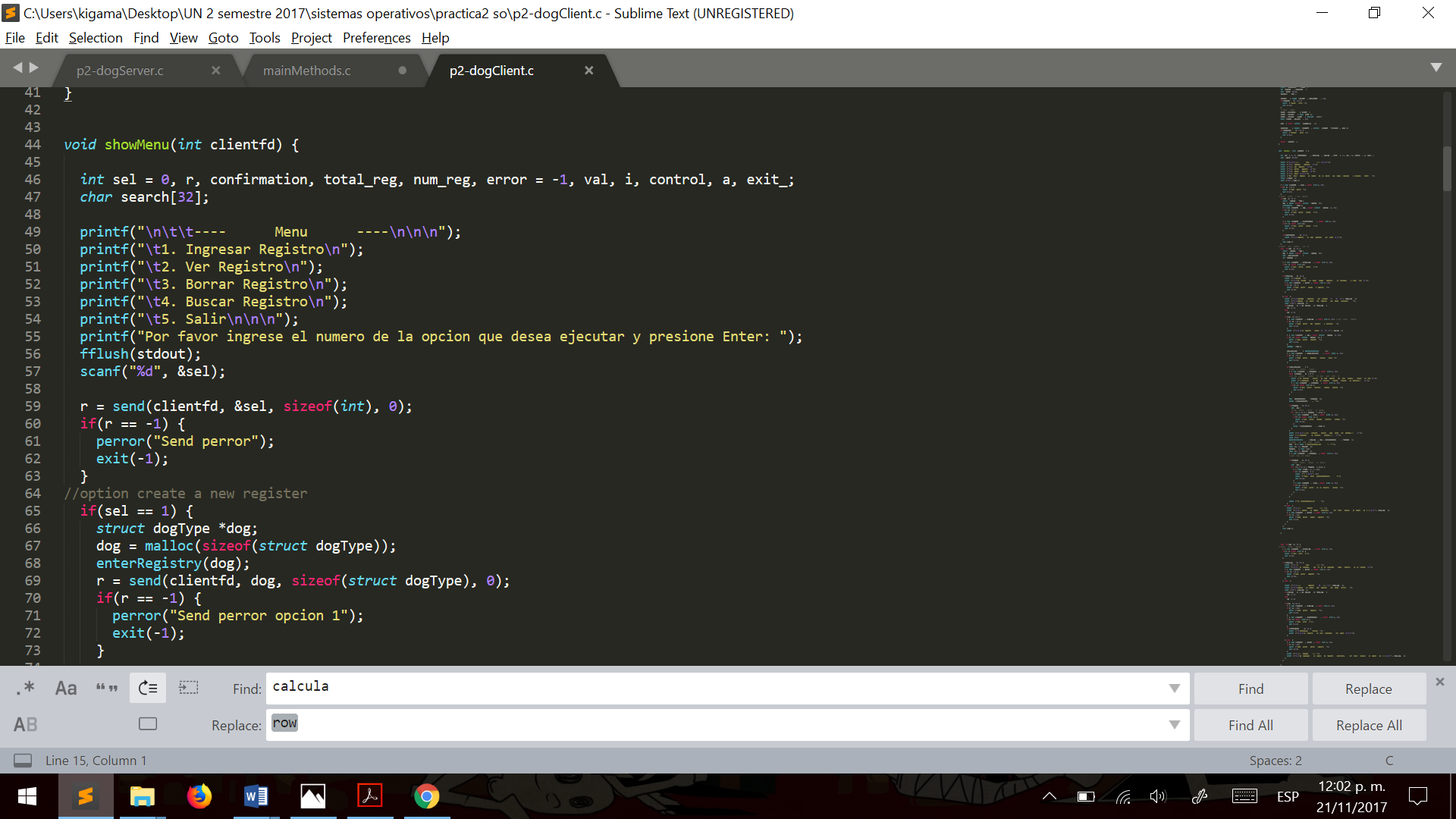
Función que busca un registro



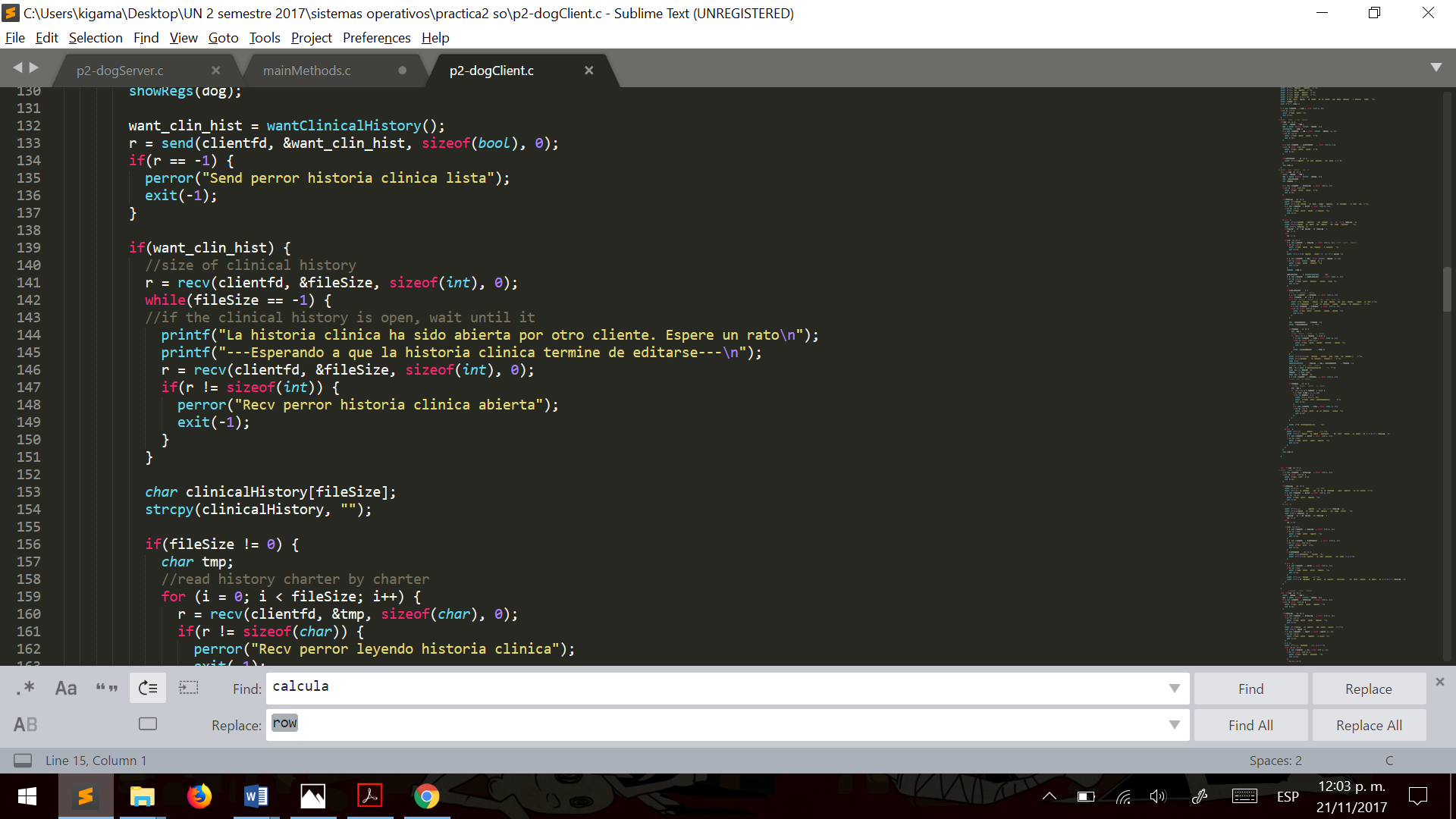
Función que retorna el tamaño del archivo



Crea una copia temporal del registro, de tal forma que copiamos todos los registros del dataDogs.dat excepto del registro que se borrara.



Código del menú del cliente



Fragmento del código donde si se encuentra abierta la historia muestra un mensaje y espera a que se desocupe.

