Informe de Elaboración

Funciones

Funciones del archivo “generator.c”.

1. generarEstructuras: Está función recibe como parametros, la cantidad de estructuras a originar, nombres, razas y tipos, estos tres últimos son para llenar los campos de cada estructura respectivos. Esta función genera y/o escoge los datos de cada campo de las estructuras aleatoriamente. Para optimizar su tiempo se escribe de a 50 mil estructuras en el archivo “dogData.dat” hasta completar la cantidad.

2. extraerDatos: tiene como parametros la ruta del archivo y el arreglo en donde se van a guardar los datos. Está función es usada para extraer los datos de los documentos “*.txt”.

3. main: La función prinicipal llama tres veces la función extraerDatos para extraer los datos de cada archivo de texto. Y almacena esos datos en tres arreglos con los respectivos datos, despúes llama a generarEstructuras para que genere las estructuras, pasandole los datos extraidos como argumentos.

Funciones del archivo “hash.c”.

1. reiniciar_hash: Esta función lee cada estructura, verifica que el valor de la entrada (valor hash), en la tabla hash, sea 0, si es cero quiere decir que ninguna estructura se ha vinculado con esta entrada, por lo que se modifica con el valor de la posición en el archivo de la primera estructura que sea vinculada a esta entrada. De lo contrario no se modifica la tabla hash.

2. hash_value: Es la función que calcula el valor hash de la estructura, tomando como parametro una cadena de texto (nombre de la mascota). Retorna el valor entero que es el hash.

3. lower_case: convierte los caracteres de una cadena de texto a minusculas.s.

Funciones del archivo “structures.c”.

1. imprimir_mascota: Imprime los campos de la estructura “dogType”. Tiene como parametro un apuntador.

2. crear_registro: Crea y pide los datos de una estructura “dogType”. Retorna un apuntador a uan estructura “dogType”.

Funciones del archivo “p2-dogClient.c”.

1. enviar: envia los datos o mensajes al servidor. Recibe como parametros un apuntador a los datos y el tamaño de los datos. Ademas valida si hubo algún error.

2. recibir: recibe los datos o mensajes del servidor. Recibe como parametros los mismos de enviar y también valida su hubo algún error.

3. buscar_registro: pide al usuario ingresar el nombre de la mascota, el cual es enviado al servidor, si la respuesta del servidor es diferente de cero, se siguen recibiendo las demas respuestas hasta que ya no haya más, es decir hasta que se reciba un cero.

4. eliminar_registro: recibe y muestra el número de registros que hay presentes en la base de datos, pide el número del registro a eliminar y se lo envía al servidor.

5. ver_registro: Recibe y muesra el número de registos que hay presentes en la base de datos, muestra, luego pide el número de registro a ver, se envía al servidor para que este responda enviando el tamaño y luego el contenido de la historia clinica, ese contenido se escribe en un archivo tempral, el cual se muestra en el editor “gedit”, se lee de nuevo el contenido del archivo y se envia al servidor y se elimina el archivo temporal.

6. insertar_registro: envia la estructura dogType, la cual es señalada a través de un apuntador como parametro.

7. mostrar_menu: Imprime las cinco acciones que se pueden realizar.

8. get_info: tiene como parametros dos cadenas de texto, y un apuntador, el primer parametro es el mensaje que se quiere imprimir, el segundo, es el formato de los datos a obtener, el apuntador, es para guardar el dato en el espacio de memoria al cual apunta.

9. menu: llama a mostrar_menu, luego pide la opción, si la opción es valida, se envía al servidor, para que esté ingrese en la misma opción. Dependiendo de la opción llama a insertar_registro, al cual le pasa como argumento la función crear_registro, o llama a ver_registro, o a eliminar_registro, o a buscar_registro, mientras la opción sea diferente de cinco. Si la opción no es valida se imprime el mensaje de advertencia y se continua pidiendo la opción.

10. crear_socket: Como el nombre indica, crea el socket del cliente y se conecta con el servidor.

11. main: llama a crear_socket y a menu.

Funciones del archivo “p2-dogServer.c”.

1. escribir_log: Escribe las peticiones que le llegan de los clientes en el archivo “serverDogs.log”. Tiene como parametros un descriptor del socket del cliente de tipo entero, un entero indicando el tipo de operación, y la solicitud que es una cadena de texto.

2. enviar: envia los datos o mensajes al cliente. Recibe como parametros un entero descriptor del socket del cliente, apuntador a los datos y el tamaño de los datos. Ademas valida si hubo algún error.

3. recibir: recibe los datos o mensajes del cliente. Recibe como parametros los mismos de ‘enviar’ y también valida su hubo algún error.

4. buscar_registro: Tiene como parametro un entero descriptor del socket del cliente. Recibe el nombre que va a buscar por parte del cliente, mira en la tabla hash si la entrada está vaciía o no, si no lo está, busca en el archivo desde la posición que está guardada en la entrada de la tabla hash los nombres que coincidan, si encuentra alguno(s), envia la posicion de la(s) coinsidencia(s), por último.

5. eliminar_registro: Tiene como parametro un entero descriptor del socket del cliente. Calcula y envia la cantidad de registros que hay al cliente, intercambia el último registro con el registro a eliminar en el archivo, se elimina la historia clinica relacionada, y se trunca el archivo en la última posición del archivo, donde ahora se encuentra el registro a eliminar, lo que disminuye el tamaño y elimina el registro de la última posición. Por último se llama a “escribir_log”.

6. enviar_historia: Tiene como parametros tiene un entero (descriptor del socket del cliente), otro valor entero (posición del registro en el archivo) y un apuntador a un fichero. Primero se comprueba que exista una historia clínica relacionada con la mascota, si no existe una, se crea y se inicializa con la informacion de la mascota. Se envía el tamaño del archivo y luego el contenido del mismo. Luego, se recibe el tamaño y el contenido del archivo, modificado o no, desde el cliente y se guarda en la base de datos, en la carpeta “historias”.

7. ver_registro: Tiene como único parametro un entero, descriptor del socket del cliente. Empieza enviando el número de registros, recibe el número del registro el cual quiere ver el cliente, luego se llama a la función “enviar_historia”. Finalmete se llama la función “escribir_log”.

8. insertar_registro: Tiene el mismo parametro que “ver_registro”. Esta función empieza recibiendo la estructura de tipo “dogType” enviada por el cliente, la escribe en el archivo “dogData.dat”, finalmente, se envia la posición del registro en el archivo como forma de confirmación con el cliente.

9. crear_socket: Como el nombre indica, crea el socket del servidor y lo pone en espera a las conexiones de los clientes, con máximo 32 conexiones.

10. atencion_cliente: Tiene un apuntador de tipo entero al descriptor del socket de un cliente como parametro. Empieza recibiendo la opción que el cliente ha escogido, para ingresar a la misma opción y atender las solicitudes que el cliente.

11. main: llama a crear_socket, luego llama a reiniciar_hash, después, mientras no se haya alcanzado el número máximo de conexiones, asigna un hilo a cada conexión establecida (descriptor del socket del cliente), el cual, la función que ejecuta cada hilo es atencion_cliente y como argumento se le pasa el apuntador al descriptor del socket del cliente.

Diagrama de Comunicación

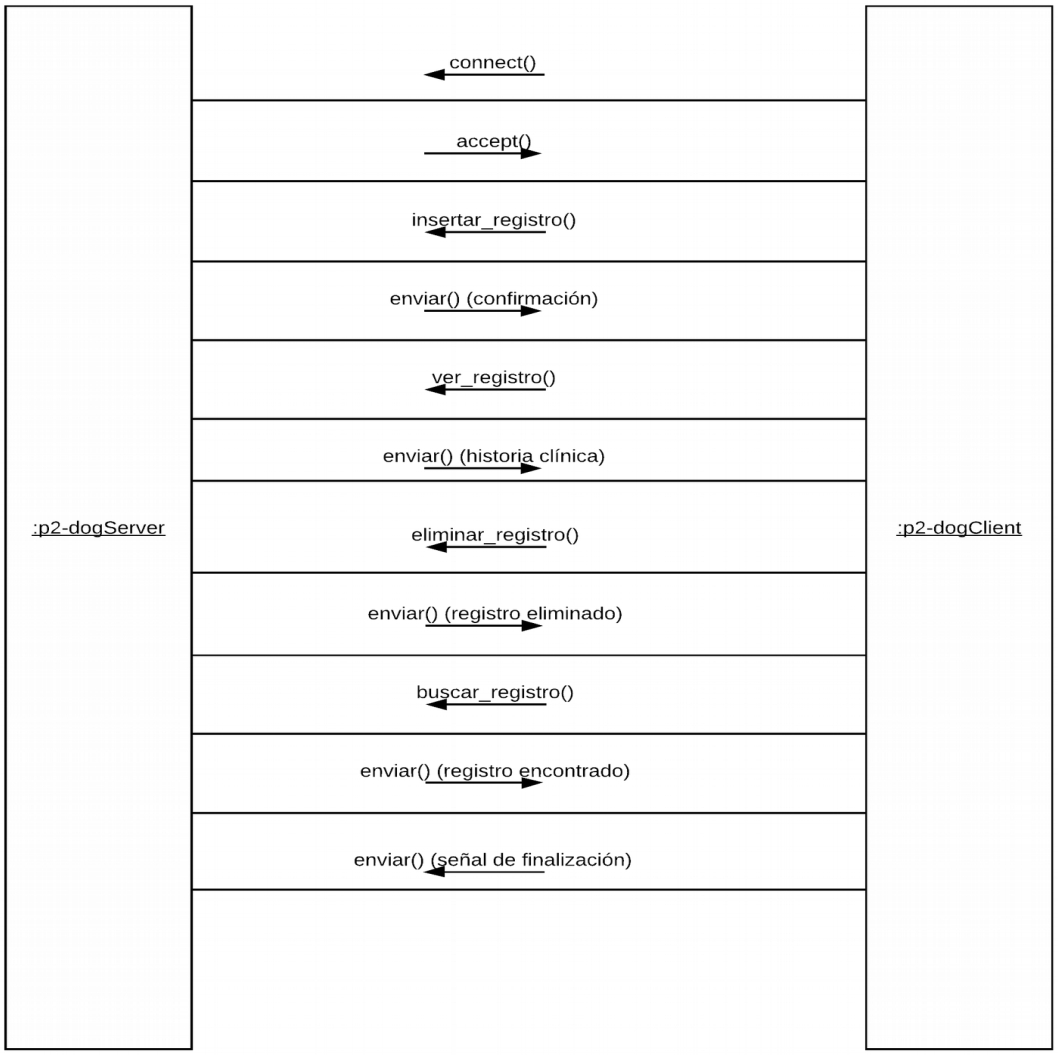

Diagrama de Bloques