**Prueba de caja blaca**

1. **Diagrama de flujo**

void volcar\_Matriculas(Matriculas \*matri){

int i=0; 1

FILE\* Matricula; 2

Matricula=fopen("Matriculas.txt","w"); 3

if(Matricula==NULL){ 4

printf("No se ha podido abrir el fichero de Matriculas. \n"); 5

exit(1); 6

}

else{

while(i<num\_lin\_Matriculas.lon){ 7

fprintf(Matricula,"%s-%s\n",matri[i].Id\_materia,matri[i].Id\_alum); 8

i++; 9

}

}

fclose(Matricula); 10

} 11

Diagrama

Descripción generada automáticamente

1. Complejidad ciclomática

VG= Aristas-nodos+2=11-11+2=2

1. Rutas
2. Ruta\_1: 1-4-5-6
3. Ruta\_2: 1-10-8-10-11
4. Pruebas de las rutas

* Ruta\_1: En esta ruta lo que tenemos que comprobar es que si el fichero de matrícula sufre algún error o deje de existir dentro del código se ejecute una serie de funciones que lo que harán será mostrar por pantalla que el fichero no se ha podido abrir y cerrar el programa. Para realizar la comprobación de esta ruta primero se deberá probar dentro del programa que el fichero de usuarios no se habrá o de un error, por lo que quitaremos la función de abrir el fichero. Cuando compilamos lo que ocurre es que al no abrir el fichero y no existir dentro de lo que es el software al llegar al punto 4 al ser matrícula igual a NULL pasará a ejecutar la instrucción del punto 5 que mostrará por pantalla el mensaje que nos indicará que ha ocurrido un problema con el fichero, después de eso se ejecutará el punto 6 que cerrará el programa debido al error, por lo que todo ocurre de la manera prevista.
* Ruta\_2: En esta ruta lo que queremos comprobar es que los datos que se almacenen de manera correcta dentro del fichero de matrícula para su posterior uso en usos posteriores del software. Para poder realizarlo usaremos la función que se encarga de volcar los datos del fichero en la estructura de datos, para luego poder comprobar que se han volcado de manera correcta al fichero. En primer lugar, en el punto 4 al ver el programa que el fichero existe y es correcto pasará al punto 7-8-9-10, al ver que la condición del bucle todavía se cumple vuelve al punto 8 de nuevo y vuelve a pasar por los puntos 9 y 10, pero esta vez al no cumplirse la condición del bucle la función terminará exitosamente, por lo que todo ocurre de la manera que teníamos prevista.

1. Prueba de los bucles

* **No se cumple la condición del bucle**:

Para comprobar que ocurre cuando no se cumple la condición del bucle solo tenemos que hacer es que la estructura de datos se encuentre vacía a la hora de volcar los datos de las estructuras a los ficheros, por lo que haremos será cambiar el valor de i e igualarlo a num\_lin\_Matriculas.lon para que no se cumpla la condición y no entre dentro del bucle. Al ocurrir esto en el fichero no se escribe nada y al estar abierto en formato de escritura se borra todo lo que había con anterioridad en él, por lo cual obtenemos un fichero vacío tal y como tenía que ocurrir y el bucle terminaría de manera correcta.

* **Una única iteración:**

Para comprobar que ocurre cuando se realiza una única interacción del bucle solo tenemos que hacer que se escriba una única línea en el fichero a la hora de volcar los datos de las estructuras a los ficheros, por lo que haremos será cambiar el valor que tiene i para que sea una unidad menos que num\_lin\_Matriculas para que la condición del bucle solo se pueda cumplir una sola vez. Al ocurrir esto en el fichero solo se escribe una única línea, que es este caso son los datos que estaban almacenados en la última posición de la estructura de datos, por lo cual obtenemos un fichero con una única línea tal y como tenía que ocurrir y el bucle termina de manera correcta.

* **Más de dos iteraciones:**

Para comprobar que ocurre cuando se realiza varías interacciones del bucle solo tenemos que hacer que se escriba n líneas en el fichero a la hora de volcar los datos de las estructuras al fichero, por lo que haremos será almacenar dentro de la estructura encargada de almacenar los datos del fichero de matriculas n líneas del fichero por lo que por cada posición dentro de la estructura se almacenará una línea del fichero. En el bucle con el contador avanzaremos posiciones dentro de la estructura mientras escribe los datos correspondientes dentro del fichero y obtenemos un fichero con n líneas de información validas, por lo que todo ocurre como tenía que ocurrir y el bucle termina de manera correcta.

**Prueba de caja negra**

Para realizar las pruebas de caja negra tendremos que usar los datos que puedan dar fallos al programa, al recibir la función la estructura y la cantidad de líneas del fichero nos centraremos en los errores provocados por recibir fallos por parte de estos parámetros, por lo que para la comprobación de lo que puede ocurrir comprobaremos una serie de casos en donde se pueden producir esta clase de errores, los cuales son:

* **La estructura tiene algunos datos vacíos:** En este caso nos referimos que al introducir los datos en el fichero alguna línea del fichero de matriculas no esta completa, por ejemplo, que en alguna línea solo estuviera el id del alumno, pero no apareciera el de la materia, “-123456”. Lo que ocurriría es que a la hora de escribir en el fichero no se produciría ningún tipo de error ya que la variable , en este caso el id de la materia, tiene el valor nulo porque no ha sido inicializada por lo que en el fichero se escribirá los datos correspondientes de la función sin producir ningún error, ya que la función únicamente se encarga de escribir los datos sin importar que estén correctos o no, ya que la parte de obtener los datos de manera correcta le corresponde a la función que realiza la operación de pasar los datos del fichero a la estructura.
* **El número de líneas del fichero es erróneo:** En este caso nos referimos a que el número de líneas que se nos pasa a la función no es correcto, de aquí podemos sacar dos posibles formas del error, la cual sería que sean menos de las que existen y en el otro caso que sean más de las que existen. En el primer caso lo que ocurriría a la hora de escribir en el fichero no se producirá ningún tipo de error y escribirá los datos correspondientes correctamente, pero faltando los datos que se han saltado. En el segundo caso lo que ocurrirá es que no se escribirá correctamente porque le estamos pidiendo que escriba un dato que no existe por lo cual en ese caso la función no funciona. En estos dos casos no se puede hacer nada para poder solucionarlo, ya que no depende de esta función calcular el número de líneas, esta solo lo recibe e imprime las líneas que le piden, por lo que en conclusión esta correcta porque la parte de calcular las líneas no forma parte de ella por lo que al llamar a esta función el número de líneas que le pasemos deberá ser correcto para que no se produzcan ningún tipo de errores y para que no se pierda ningún dato.

Prueba de Calificaciones\_matriculas

En este apartado se documentarán varias pruebas realizadas con el módulo correspondiente para comprobar su correcto funcionamiento. Para este módulo al ser uno que esta lleno de funciones que realizan funciones independientes de las otras, en su mayoría, pasaremos a analizar cada función por separado y aquellas que se complementa las realizaremos juntas.

* **introducir\_Matriculas:** En esta función su principal función es que se introduzcan correctamente los datos guardados en el fichero en la estructura de datos correspondientes para poder usarlos de manera correcta dentro del programa. Para la prueba lo que haremos será guardar dentro del fichero la siguiente línea,” 0001-878722”. Al iniciar la función lo primero que hace es crear el fichero e inicializar el dato de tipo char, una cadena de caracteres, poner el número de líneas a 0 y la estructura de datos darle un valor nulo. Ahora se nos abrirá el fichero de matriculas y como el fichero existe no nos dará ningún problema, por lo que el programa pasa a meter los datos dentro de la estructura. En primer lugar, la primera línea del fichero se guardará la la variable línea, y como existe una línea de texto entramos dentro de la condición y con un realloc creamos una nueva posición para la estructura de datos para almacenar los datos de la línea seleccionada. Ahora guardados dentro de la variable len el número de caracteres dentro de la línea y le quitamos el “\0” para que no existan problemas. La función strtok guarda dentro de la variable token la parte de la línea hasta el símbolo “- “y si no es nulo lo almacena en (\*matri)[\*n].Id\_materia. Después realiza de nuevo la misma función y guarda hasta el siguiente “- “o el final de la línea y lo almacena dentro de (\*matri)[\*n].Id\_alum. Por último, suma uno al valor de n y volvemos al principio del bucle, pero como ya no ahí mas línea salimos y se nos cierra el fichero tiendo la información de este perfectamente guardada dentro de la estructura de datos para las matrículas.
* **introducir\_Calificaciones:** En este caso es la misma función que la anterior y lo único distinto es que en vez de usar el fichero encargado de almacenar los datos correspondientes a las matriculas usamos el que corresponde con los datos almacenados de las calificaciones. Por lo que la prueba sería exactamente igual a la de la anterior función, variando únicamente que en el fichero de calificaciones existen más campos que almacenar que la anterior, pero sigue exactamente el mismo proceso que el anterior para su prueba. (Para la prueba usaremos la siguiente línea en el fichero: 25/02/2017-Examen unidad 1-0001-342312-7)
* **volcar\_Matriculas y volcar\_Calificaciones:** En este caso realizaremos la prueba de los dos al mismo tiempo por las funciones son idénticas, lo único que cambia es el fichero que abren y la cantidad de cosas que escriben por lo que se harán las dos pruebas al mismo tiempo. Al principio ponemos un contador a 0 en los dos y abrimos el fichero correspondiente y en ninguno de los dos se produce ningún tipo de error que no permita a la función continuar, por lo que entramos en un bucle donde se escribirá línea por línea toda la información almacenada dentro de la estructura de datos correspondiente, en este caso (Se han usado para las pruebas los mismos datos de los dos apartados anteriores) solo se escribe dentro de los ficheros una única línea de código por lo que la salida de los dos era correcta, ya que para los dos solo le introducimos al programa un única línea de código.
* **materias\_alum:** En este caso le vamos a pasar a esta función la estructura que tiene almacenados los datos de las matrículas y el id de un alumno, como usaremos los datos anteriores le pasaremos el id “878722” para que se nos imprima por pantalla las materias en las que se encuentra matriculado. Entonces se incia dos contadores (i y cont) y entramos dentro de un bucle que avanza posiciones dentro de la estructura de datos para encontrar alguna matricula que tenga un alumno matriculado con dicho id. Como tienen el mismo id se muestra por pantalla la asignatura en la que se encuentra matriculado y como no existen más datos se termina la función.
* **Imprimir\_nota:** En este caso le vamos a pasar a la función la estructura de datos encargada de almacenar los datos sobre las calificaciones, un id de un alumno existente y el id de una materia, en este caso será 342312-0001. En primer lugar en la función se inicializa dos contadores (i y cont) y entramos dentro de un bucle donde recorremos toda la estructura de datos donde se almacenan la calificaciones. Dentro del bucle primero comprobamos que el id que hemos pasado existe, en este caso nos permite pasar, ya que existe. Después pasamos a otra condición que comprueba si el id de la materia del alumno que tenemos concuerda con el id de materia de la nota, como nos coincide nos muestra por pantalla la nota del alumno en dicha asignatura, por lo que el bucle avanza a la siguiente posición, pero como nos tenemos más datos almacenados termina y pasa a la ultima condición en la cual no entramos ya que hemos encontrado que el alumno tiene alguna nota la asignatura que buscamos.
* **modificar\_nota:** En esta función le pasamos como parámetro a la función la estructura de datos donde se almacenan los datos correspondientes a las calificaciones de los alumnos, un id de un alumno y el id de una materia (el id del alumno y de la materia son los mismo que en el caso anterior). En primer lugar, lo que se hace es inicializar un contador i para el bucle for y una variable x con el valor 1. Después pasamos al bucle donde recorreremos toda la estructura de datos de las calificaciones para encontrar la nota que queremos modificar. En primer lugar, comparamos si el id del alumno tiene alguna nota almacenada y como si la tenemos pasamanos a la siguiente condición, la cual comprueba si el alumno tiene alguna nota almacenada de la asignatura que le hemos pasado, lo cual es cierto en este caso. Por último, te muestra un mensaje por pantalla de la nota que queremos modificar que el de la asignatura que le hemos pasado, es decir “Examen unidad 1” por lo que introducimos un 0 por teclado y ahora no pide una nueva nota la cual va a ser un 9. Como no existen más notas dentro de la estructura de datos de calificaciones la función termina.
* **eliminar\_nota:** En esta función le pasamos como parámetro a la función la estructura de datos donde se almacenan los datos correspondientes a las calificaciones de los alumnos, un id de un alumno y el id de una materia (el id del alumno y de la materia son los mismo que en el caso anterior). En primer lugar, lo que se hace es inicializar un contador i para el bucle for y una variable x con el valor 1. Después pasamos al bucle donde recorreremos toda la estructura de datos de las calificaciones para encontrar la nota que queremos eliminar. En primer lugar, comparamos si el id del alumno tiene alguna nota almacenada y como si la tenemos pasamanos a la siguiente condición, la cual comprueba si el alumno tiene alguna nota almacenada de la asignatura que le hemos pasado, lo cual es cierto en este caso. Después de comprobar que existe dicha calificación se nos mostrará un mensaje por pantalla que nos preguntará si deseamos eliminar dicha calificación por lo cual como queremos eliminarla pulsaremos el número 1 tal y como se nos indica por pantalla. Entremos dentro de la siguiente condición y llamamos a una función que se llama adelantar\_Calificaciones, a la cual le pasamos como parámetro la estructura de datos en donde se almacenan los datos sobre las calificaciones y la posición la cual deseamos eliminar. Dentro de la función inicializamos un contador en la posición pasada por parámetros y empezamos a copiar el contenido de la siguiente posición en la que queremos eliminar, como la siguiente posición no existe se pondrá todo al nulo, ya que solo tuvimos una posición en la estructura de datos. La función termina y volvemos a la función de eliminar nota, en la cual realizamos un realloc para eliminar una posición dentro de la estructura de datos y el contador de líneas lo reducimos en una unidad. La función termina y al abrir el fichero nos encontramos con un fichero vacío, tal y como debería ser.
* **anadir\_nota:** En esta función le pasamos como parámetro a la función la estructura de datos donde se almacenan los datos correspondientes a las calificaciones de los alumnos, el número de líneas que han sido guardadas dentro de la estructura, un id de un alumno y el id de una materia (el id del alumno y de la materia son los mismo que en el caso anterior). En primer lugar, lo que se hace es inicializar una variable y e realizamos un realloc a la estructura de datos, ya que vamos a necesitar para almacenar los datos que vamos a añadir una nueva posición dentro de la estructura de datos. Después entramos dentro de un bucle se nos pide introducir una breve descripción para el examen y la guardamos dentro de la nueva posición, pero antes de salir del bucle se comprueba que esta no supere el límite de 32 caracteres, en mi caso yo he puesto “Examen Unidad 2” que son 15 caracteres por lo que el programa me permite continuar. Después se nos pide que introduzcamos varios datos por teclado, que son el día, mes y año, en mi caso introduje la fecha “10/04/2022”. Después entramos dentro de otro bucle en el cual nos pide que metamos por pantalla la nota que queremos en el examen, en mi caso será “6,75”. Por último, en la condición del bucle se comprueba que la nota está entre 0 y 10 y como la mía lo cumple sale del bucle, aumenta en 1 el número de líneas almacenado en la estructura e imprime un mensaje por pantalla para mostrar de que se ha guardado la nueva calificación, la cual al abrir el fichero nos encontramos con “10/04/2022- Examen Unidad 2- 0001-342312-6´75”, lo cual es lo que hemos introducido.
* **anadir\_matricula:** En esta función le pasamos como parámetro la estructura de datos que tiene almacenados los datos correspondientes con las matrículas, el id de un alumno y el id de una materia, en este caso los datos que le pasamos serán “0002-123456”.En primer lugar hacemos un realloc a la estructura de datos para poder almacenar los datos correspondientes a la nueva matrícula que vamos a hacer y después copiamos los datos que hemos introducido por parámetros a las nuevas posiciones, aumentamos en 1 el número de líneas que se encuentran almacenadas dentro de la estructura y se muestra por pantalla un mensaje para decir que se han almacenado con éxito los datos. Por último, abrimos el fichero y observamos que se ha añadido una nueva línea que es “0002-123456”.
* **borrar\_matricula:** En esta función le pasamos como parámetro la estructura de datos que tiene almacenados los datos correspondientes con las matrículas, el id de un alumno y el id de una materia, en este caso los datos que le pasamos serán “0002-123456” (usaremos el mismo fichero que en la prueba anterior). En primer lugar, lo que hacemos es inicializar una variable i que nos servirá de contador y entrar en un bucle en donde recorreremos todas las posiciones dentro de la estructura de datos. Dentro del bucle no encontraremos con una condición que en la cual podremos entrar si el id del alumno coincide con el id de la posición y si el id de la materia coincide con id de la materia. En la primera posición no entrará por lo que avanza una posición, pero en este si consigue entrar, por lo que se pasa a llamar a la función adelantar\_Matriculas. En dicha función pasamos por parámetro la estructura de datos donde tenemos almacenados los datos correspondientes a las matriculas y la posición de la estructura de datos que queremos eliminar. Después pasamos al bucle donde se adelantarán los contenidos de los datos en una posición, pero con este caso no existe la siguiente posición pasarán a tener un valor nulo, por lo que terminado esto se volverá a la anterior función. Después, se hace un realloc para eliminar la última posición de la estructura y el contador de líneas lo reducimos en una posición. Por último, llamamos a la función de eliminar\_nota y se eliminan las notas de las asignaturas que ya no existen en el sistema.
* **modificar\_matricula:** En este caso la función le pasamos como parámetros la estructura que almacena los datos sobre las matrículas, el id de un alumno, el id de una materia vieja y el id de una materia de la que se desea matricular, los datos que ser pasarán a la función serán “123456-0002-0101”. En primer lugar, inicializamos una variable i para que sea un contador y entramos dentro del bucle, dentro recorreremos toda la estructura de los datos de las matrículas y con la condición del if como la primer y única posición tiene el mismo id de alumno y el mismo id de la aignatura vieja nos permite modificar ese id por el de la nueva asignatura de la que nos queremos matriculas. Como solo tiene una posición termina el bucle y se llama a la función de eliminar\_nota, y por último la función termina y al abrir el fichero nos damos cuenta de que el id de la asignatura a cambiando al nuevo.
* **eliminar\_matricula\_materia:** En este caso a la función le pasamos como parámetro la estructura de datos que almacena los datos correspondientes a la matrícula y el id de una materia, el cual será “0002” (para esta prueba usaremos un fichero donde solo estará una línea de código, la cual es”0002-111222”). En primer lugar, inicializamos un contador y entramos dentro del bucle donde mediante una condición se comprueba si el id de la materia que hemos pasado coincide con alguno de los que tienen almacenados, en este caso el primer coincide con el que le hemos pasado por lo que pasamos a llamar a la función de adelantar matriculas, la cual adelanta las posiciones de la estructura copiando los datos de las i+1 posiciones en las posiciones i. Después de que se termine la función hacemos un realloc, el cual eliminará la última posición de la estructura de datos y reduciremos el número de líneas almacenadas en 1. Se nos muestra un mensaje por pantalla diciéndonos que se han eliminado las matrículas de dicha asignatura correctamente y al abrir el fichero nos encontramos con que está vacío, que es el resultado esperado.
* **eliminar\_matriculas\_alumno:** En este caso a la función le pasamos como parámetro la estructura de datos que almacena los datos correspondientes a la matrícula y el id de un alumno, el cual será “111222” (para esta prueba usaremos un fichero donde solo estará una línea de código, la cual es”0002-111222”). En primer lugar, inicializamos un contador y entramos dentro del bucle donde mediante una condición se comprueba si el id del alumno que hemos pasado coincide con alguno de los que tienen almacenados, en este caso el primer adelantar matriculas, la cual adelanta las posiciones de la estructura copiando los datos de las i+1 posiciones en las posiciones i. Después de que se termine la función hacemos un realloc, el cual eliminará la última posición de la estructura de datos y reduciremos el número de líneas almacenadas en 1. Se nos muestra un mensaje por pantalla diciéndonos que se han eliminado las matrículas del dicho alumno correctamente y al abrir el fichero nos encontramos con que está vacío, que es el resultado esperado.
* **eliminar\_calificaciones\_materia:** En este caso a la función le pasamos como parámetro la estructura de datos que almacena los datos correspondientes a las calificaciones y el id de un alumno, el cual será “342312” (para esta prueba usaremos un fichero donde solo estará una línea de código, la cual es” 10/04/2022- Examen Unidad 2- 0001-342312-6´75”). En primer lugar, inicializamos un contador y entramos dentro del bucle donde mediante una condición se comprueba si el id del alumno que hemos pasado coincide con alguno de los que tienen almacenados, en este caso el primer adelantar calificaciones, la cual adelanta las posiciones de la estructura copiando los datos de las i+1 posiciones en las posiciones i. Después de que se termine la función hacemos un realloc, el cual eliminará la última posición de la estructura de datos y reduciremos el número de líneas almacenadas en 1. Se nos muestra un mensaje por pantalla diciéndonos que se han eliminado las calificaciones del dicho alumno correctamente y al abrir el fichero nos encontramos con que está vacío, que es el resultado esperado
* **eliminar\_calificaciones\_alumnos:** En este caso a la función le pasamos como parámetro la estructura de datos que almacena los datos correspondientes a las calificaciones y el id de una materia, el cual será “0001” (para esta prueba usaremos un fichero donde solo estará una línea de código, la cual es” 10/04/2022- Examen Unidad 2- 0001-342312-6´75”). En primer lugar, inicializamos un contador y entramos dentro del bucle donde mediante una condición se comprueba si el id de la materia que hemos pasado coincide con alguno de los que tienen almacenados, en este caso el primer coincide con el que le hemos pasado por lo que pasamos a llamar a la función de adelantar calificaciones, la cual adelanta las posiciones de la estructura copiando los datos de las i+1 posiciones en las posiciones i. Después de que se termine la función hacemos un realloc, el cual eliminará la última posición de la estructura de datos y reduciremos el número de líneas almacenadas en 1. Se nos muestra un mensaje por pantalla diciéndonos que se han eliminado las calificaciones de dicha asignatura correctamente y al abrir el fichero nos encontramos con que está vacío, que es el resultado esperado.