

DOCUMENTACIÓN EXTERNA CDP

Este programa contiene los siguientes módulos c con sus correspondientes módulos h:

- admin.c, admin.h, menuadmin.c, menuadmin.h, tipos.h, volfichest.c, volfichest.h, volestfich.c, volestfich.h, Profesor.c, Profesor.h, main.c

La funcionalidad que tiene este programa es la de solucionar el problema planteado, el de crear un cuaderno práctico para el profesorado actual. Con este el usuario, cuyo papel será el de profesor/a o el de administrador/a, podrá acceder al sistema con una cuenta formada por una ID propia y una contraseña seleccionada por él/ella mismo/a. Tras haber accedido al sistema, podrá elegir distintas opciones dependiendo de su perfil. Si es profesor podrá tener acceso a los datos, entre muchos otros, de los alumnos, sus calificaciones, sus faltas de asistencia etc. Y si es administrador, se encargará de todo lo que tenga que ver con la modificación de esos datos, como por ejemplo dar de alta a un alumno de nuevo ingreso o hacer el cambio de horario a un miembro del profesorado.

Este problema ha sido dividido en distintos módulos para facilitar su corrección y legibilidad. La división principal consta de un módulo de volcado de información de los ficheros a las estructuras creadas en el módulo “tipo.h”, el programa principal compuesto por distintos módulos que próximamente mencionaremos y comentaremos y ,por último un módulo que guarde la información modificada de las estructuras de vuelta a los ficheros. Los módulos que componen el programa principal son los siguientes: admin, profesor, tipos, login, menus, volestfich y volfichest.

El módulo admin tiene como función encargarse de todas las tareas que debe poder realizar el usuario que sea “administrador”. Este tiene distintas funciones con objetivos diferenciados, por un lado hay algunas que se encargan de cualquier modificación en los datos de los distintos usuarios como dar de baja, alta o modificar alguno de ellos. Por otro, las demás se encargarán de la información perteneciente a los alumnos, materias y horarios respectivamente.

El módulo Profesor tiene como función encargarse de todas las tareas que debe poder realizar un usuario con perfil ”Profesor”. Tiene funciones de distinta naturaleza como listar las faltas de un alumno dado, modificarlas e incluso poder añadir nuevas o eliminar ya existentes. Puede de la misma manera realizar otras actividades como por ejemplo, listar los datos personales del alumno y poder modificarlos etc.

El módulo login tiene como función encargarse del acceso de los usuarios al sistema, tanto usuarios con perfil “profesor” o “administrador”. Este pide al usuario su ID y su contraseña para el acceso a sus funciones exclusivas de modificación o visualización.

El módulo tipos tiene como función la de contener todas las estructuras (con la información de ficheros almacenada previamente) y declaraciones que van a ser usadas en los demás módulos.

El módulo menuadmin contendrá todos los menus en los que cada usuario (dependiendo de si tiene perfil “administrador” o “profesor” tienen unos menus u otros) tendrá la opción de seleccionar entre distintas funciones relacionadas con su perfil.

El módulo volfichest se encarga de cargar en el programa la información procedente de los ficheros. Esta información se almacenará en las respectivas estructuras que se encuentran en el módulo “tipos”.

El módulo volestfich se encarga de guardar la información de las estructuras que ha sido modificada durante la ejecución del programa y a través de varias funciones de nuevo en los ficheros.

Antes de utilizar el programa debe tener instalado un programa cuya función sea la del desarrollo de programas en c (como por ejemplo Code Blocks o Dev C++). Para poder utilizar este programa debe primero descomprimir el archivo que lo contiene y más tarde ejecutarlo con alguno de los programas de ejemplo. Después de haberlo ejecutado debe pulsar en la pestaña Ejecutar que se encuentra en la parte superior y de nuevo a Ejecutar. Ya podrá hacer uso libre del programa en el caso de que quiera ser utilizado.

PRUEBAS

Empezamos realizando las pruebas de unidad, que consisten en comprobar cada módulo por separado. Comenzamos con el módulo menuadmin.

En primer lugar, realizamos las pruebas de caja blanca. Para ello planificamos un conjunto de rutas de ejecución que nos permiten recorrer cada instrucción al menos una vez eficientemente. Esas rutas se calculan mediante una serie de pasos que buscan determinar la complejidad ciclomática y el conjunto de rutas independientes. Además se calculan las iteraciones máximas de cada bucle.

Tras realizar estos cálculos, comenzamos con las pruebas. Tras recorrer cada ruta, comprobamos que no ha habido errores, y que cada instrucción ha reaccionado de la forma prevista. En cada bucle, se ha comprobado la reacción al no cumplirse la condición, con una sola iteración, con dos, en definitiva con cualquier número por debajo del máximo, así como con el máximo y con un número por encima.

En los while, como se esperaba, todos los números de fuera del margen simplemente nos devolvían al propio bucle, y las de dentro daban el resultado esperado. En el caso de los switch, escoger una opción fuera del bucle provoca la salida de la función correspondiente, como era de esperar, mientras que escoger una dentro del mismo realiza la instrucción debida. Los bucles for funcionaron como era de esperar y sin imprevistos.

Tras esas comprobaciones, damos comienzo a las pruebas de caja negra, en las cuales debemos buscar funciones incorrectas o ausentes, mediante la introducción de datos que afecten a los extremos y a los primeros valores que los sobrepasan (el inferior al mínimo y el superior al máximo)

Con estas pruebas, se detectó que la falta de una instrucción default en los bucles switch provocaba el final del programa al escoger una opción inválida. No se detectó errores o deficiencias más allá de eso.

A continuación, probamos el módulo login, comenzando con las pruebas de caja blanca, repitiendo los procesos anteriores.

Los bucles for funcionaron como era de esperar y sin imprevistos, y los if funcionan correctamente e incorporan un else en caso de error.

A continuación, realizamos las pruebas de caja negra. En ellas, en este caso, no detectamos ningún error o deficiencia.

A continuación, probamos el módulo admin, comenzando con las pruebas de caja blanca, repitiendo los procesos anteriores.

Los bucles for funcionaron como era de esperar y sin imprevistos, y los if funcionan correctamente e incorporan un else en caso de error.

En los while y do-while, como se esperaba, todos los números de fuera del margen simplemente nos devolvían al propio bucle, y las de dentro daban el resultado esperado.

Tras ello, realizamos las pruebas de caja negra. En este caso no detectamos ningún error o deficiencia.

A continuación, probamos el módulo Profesor, comenzando con las pruebas de caja blanca, repitiendo los procesos anteriores.

Los bucles for funcionaron como era de esperar y sin imprevistos, y los if funcionan correctamente e incorporan un else en caso de error.

En los while y do-while, como se esperaba, todos los números de fuera del margen simplemente nos devolvían al propio bucle, y las de dentro daban el resultado esperado.

Tras ello, realizamos las pruebas de caja negra. En este caso no detectamos ningún error o deficiencia.

A continuación, probamos el módulo `volestfich`, comenzando con las pruebas de caja blanca, repitiendo los procesos anteriores.

Los bucles `if` funcionan correctamente e incorporan un `else` en caso de error.

En los `while`, como se esperaba, todos los números de fuera del margen simplemente nos devolvían al bucle, y las de dentro daban el resultado esperado.

Tras ello, realizamos las pruebas de caja negra. En este caso no detectamos ningún error o deficiencia.

A continuación, probamos el módulo `volfichet`, comenzando con las pruebas de caja blanca, repitiendo los procesos anteriores.

Los bucles `if` funcionan correctamente e incorporan un `else` en caso de error.

En los `while`, como se esperaba, todos los números de fuera del margen simplemente nos devolvían al bucle, y las de dentro daban el resultado esperado.

Tras ello, realizamos las pruebas de caja negra. En este caso no detectamos ningún error o deficiencia.

Una vez probados los módulos por separado, comenzamos las pruebas de integración, es decir, comprobar que cada módulo sigue funcionando correctamente una vez integrado. Para ello, repetimos las pruebas.

Tras haber llevado a cabo la repetición, podemos asegurar que todos siguen funcionando correctamente, excepto un error que aparece al ejecutar el programa en sí. Tras ello, deberían realizarse las pruebas de validación, comprobando que el software completo sigue funcionando, pero el error “`makefile error`” lo hace imposible.