

Documentación CPD

Jesús Rodríguez Heras
Elías Makdad Khamlichi
Juan de la Cruz Caravaca Guerrero
Francisco Javier Gómez Coronil

11 de mayo de 2017

Índice general

| | |
|--|-----------|
| 1. Documentación | 5 |
| 1.1. Descripción funcional | 5 |
| 1.2. Descomposición del problema | 5 |
| 1.3. Documentación por módulo | 5 |
| 1.3.1. Módulo 1: Ficheros | 5 |
| 1.3.2. Módulo 2: Administrador | 6 |
| 1.3.3. Módulo 3: Profesor | 6 |
| 1.4. Proceso de instalación | 6 |
| 1.5. Ejecución del programa | 6 |
| 1.5.1. Perfil administrador | 6 |
| 1.5.2. Perfil profesor | 9 |
| 2. Pruebas | 11 |
| 2.1. Introducción | 11 |
| 2.2. Pruebas de caja blanca | 13 |
| 2.2.1. Prueba de la ruta básica | 13 |
| 2.3. Pruebas de caja negra | 17 |

Capítulo 1

Documentación

1.1. Descripción funcional

Este programa tiene la finalidad de ser usado para administrar un centro educativo, el cual tiene una serie de administradores (director o jefe de estudios), profesores y alumnos. De todos ellos se guardará la información más relevante para el sistema en distintos ficheros que finalmente configurarán una base de datos en la cual tendremos concentrado a todo el personal del centro.

Al iniciar el programa se realiza el volcado de memoria de los ficheros, guardados en disco, a las estructuras creadas por el programa en memoria principal, lo que deja estructurado el sistema en dicha memoria principal.

Después de realizar los cambios necesarios en el programa, este vuelve a guardar los datos existentes en la memoria principal en los ficheros del disco.

1.2. Descomposición del problema

El problema (programa) ha sido descompuesto en 3 partes bien diferenciadas tales como es el sistema de volcado de memoria, el perfil administrador y el perfil profesor.

Cada uno de las partes ha sido implementada en un módulo distinto y todo se ha enlazado de forma que haya una mayor cohesión en el módulo y el menor acoplamiento posible entre los mismos.

1.3. Documentación por módulo

1.3.1. Módulo 1: Ficheros

Este módulo se encarga del volcado de datos del disco a memoria principal y viceversa (al terminar el programa).

Contiene las estructuras (registros) mediante los cuales podremos identificar cada campo de los usuarios pedidos en el sistema.

También incorpora la funcionalidad de acceso al sistema que identifica a un usuario como administrador o profesor y les muestra sus funcionalidades disponibles.

Está implementado con una serie de funciones declaradas como `static` (para que solo puedan ser utilizadas en este módulo) y contiene unas variables globales que contabilizarán el número actual de registros presentes en el sistema.

Además, guarda una variable global con el identificador de usuario actual en el sistema para que, por ejemplo, no se pueda eliminar a sí mismo.

Estas variables globales nos darán información sobre cuantos usuarios, alumnos, materias, matrículas, calificaciones, faltas y horarios hay presentes en el sistema.

1.3.2. Módulo 2: Administrador

Este módulo se encarga de gestionar todas las funciones disponibles para un administrador. Está compuesto por una serie de funciones declaradas como `static` que se encargan de todo el proceso.

1.3.3. Módulo 3: Profesor

Este módulo se encarga de gestionar todas las funciones disponibles para un profesor. Está compuesto por una serie de funciones declaradas como `static` que se encargan de todo el proceso.

1.4. Proceso de instalación

El proceso de instalación consiste en tener el programa, ubicarlo en la carpeta deseada y abrirlo con doble click. Una vez hecho esto, el programa generará los archivos necesarios para su ejecución tales como los ficheros que deberán ser completados a mano por el usuario una vez se haya ejecutado dicho programa.

1.5. Ejecución del programa

Una vez ejecutado el programa se realiza el volcado de información de los ficheros, guardados en disco, a las estructuras creadas por el programa en memoria principal, lo que deja estructurado el sistema en la memoria principal para su uso completo.

A continuación, se pide al usuario que se identifique con su nombre de usuario (no es su nombre real, aunque también estará en el sistema) y su contraseña.

Una vez identificado, encontramos dos perfiles posibles con distintas funciones:

1.5.1. Perfil administrador

Al usuario que ha sido identificado como administrador en el sistema se le mostrará el menú principal de administrador en pantalla:

- 1.- Usuarios.
- 2.- Alumnos.
- 3.- Materias.
- 4.- Horarios.
- 5.- Salir.

1.- Usuarios

Si el administrador elige esta opción se le mostrará en pantalla el siguiente menú:

- 1.- Dar de alta un usuario.
- 2.- Dar de baja un usuario.
- 3.- Modificar un usuario existente.
- 4.- Mostrar los usuarios existentes en el sistema.
- 5.- Volver al menú anterior.

1. **Dar de alta un usuario:** En esta opción, el administrador podrá registrar un nuevo usuario en el sistema e identificarlo como administrador o profesor. Si intenta usar un identificador de usuario ya existente en el sistema, el programa mostrará un error y le pedirá que vuelva a introducir un identificador que no esté en uso.

2. **Dar de baja un usuario:** En esta opción, el administrador podrá eliminar un usuario del sistema, previa confirmación en el caso de equivocarse. Si el usuario al que intenta dar de baja es él mismo, el sistema dará un error por pantalla y le pedirá que vuelva a introducir un identificador de usuario válido.
3. **Modificar un usuario existente:** En esta opción, el administrador podrá modificar cualquier campo de cualquier usuario (tanto administradores como profesores).
4. **Mostrar los usuarios existentes en el sistema:** En esta opción se muestran todos los usuarios (administradores y profesores) que se encuentran en el sistema con todos sus campos correspondientes.
5. **Volver al menú anterior:** Esta opción devuelve al administrador al menú principal de administrador.

2.- Alumnos

Si el administrador elige esta opción se le mostrará en pantalla el siguiente menú:

- 1.- Dar de alta un alumno.
- 2.- Dar de baja un alumno.
- 3.- Modificar un alumno existente.
- 4.- Mostrar los alumnos existentes en el sistema.
- 5.- Gestionar las matriculas de un alumno.
- 6.- Volver al menú anterior.

1. **Dar de alta un alumno:** En esta opción, el administrador podrá registrar un nuevo alumno en el sistema. Si intenta usar un identificador de alumno ya existente en el sistema, el programa mostrará un error y le pedirá que vuelva a introducir un identificador que no esté en uso.
2. **Dar de baja un alumno:** En esta opción, el administrador podrá eliminar un alumno del sistema, previa confirmación en el caso de equivocarse.
3. **Modificar un alumno existente:** En esta opción, el administrador podrá modificar cualquier campo de cualquier alumno.
4. **Mostrar los alumnos existentes en el sistema:** En esta opción se muestran todos los alumnos que se encuentran en el sistema con todos sus campos correspondientes.
5. **Gestionar las matriculas de un alumno:** Si el administrador selecciona esta opción, podrá gestionar las matrículas de los alumnos, así como añadir nuevas matriculas o eliminar matrículas. Al seleccionar esta opción, se mostrará en pantalla el siguiente menú:
 - 1.- Matricular un alumno de una nueva asignatura.
 - 2.- Eliminar matrícula de una asignatura de un alumno.
 - 3.- Realizar cambios en una matrícula existente de un alumno.
 - 4.- Mostrar la matrícula de un alumno.
 - 5.- Volver al menú anterior.

- a) **1.- Matricular un alumno de una nueva asignatura:** Si se selecciona esta opción, el administrador podrá añadir una nueva matrícula de un alumno a una asignatura en la que no se encuentre matriculado.
 - b) **2.- Eliminar matrícula de una asignatura de un alumno:** En esta opción se elimina una matrícula de un alumno en la asignatura que se seleccione.
 - c) **3.- Realizar cambios en una matrícula existente de un alumno:** En esta opción se podrá modificar cualquier campo de la matrícula de un alumno.
 - d) **4.- Mostrar la matrícula de un alumno:** En esta opción el administrador podrá listar las matrículas de un alumno.
 - e) **5.- Volver al menú anterior:** Esta opción devuelve al administrador al menú de la gestión de alumnos.
6. **Volver al menú anterior:** Esta opción devuelve al administrador al menú principal de administrador.

3.- Materias

Si el administrador elige esta opción se le mostrará en pantalla el siguiente menú:

- 1.- Dar de alta una materia.
- 2.- Dar de baja una materia.
- 3.- Modificar una materia existente.
- 4.- Mostrar las materias existentes en el sistema.
- 5.- Volver al menú anterior.

1. **Dar de alta una materia:** En esta opción, el administrador podrá registrar una nueva materia en el sistema. Si intenta usar un identificador de materia ya existente en el sistema, el programa mostrará un error y le pedirá que vuelva a introducir un identificador que no esté en uso.
2. **Dar de baja una materia:** En esta opción, el administrador podrá eliminar una materia del sistema, previa confirmación en el caso de equivocarse.
3. **Modificar una materia existente:** En esta opción, el administrador podrá modificar cualquier campo de cualquier materia.
4. **Mostrar las materias existentes en el sistema:** En esta opción se muestran todas las materias que se encuentran en el sistema con todos sus campos correspondientes.
5. **Volver al menú anterior:** Esta opción devuelve al administrador al menú principal de administrador.

4.- Horarios

Si el administrador elige esta opción se le mostrará en pantalla el siguiente menú:

- 1.- Añadir horas de clase.
- 2.- Eliminar horas de clase.
- 3.- Modificar horas de clase.
- 4.- Mostrar los horarios existentes en el sistema.
- 5.- Volver al menú anterior.

1. **Añadir horas de clase:** En esta opción, el administrador podrá añadirle horas de clase al profesor que elija.
2. **Eliminar horas de clase:** En esta opción, el administrador podrá eliminar horas de clase al profesor que elija.
3. **Modificar horas de clase:** En esta opción, el administrador podrá modificar las horas de clase al profesor que elija.
4. **Mostrar los horarios existentes en el sistema:** En esta opción se muestran todos los horarios que se encuentran en el sistema con todos sus campos correspondientes.
5. **Volver al menú anterior:** Esta opción devuelve al administrador al menú principal de administrador.

5.- Salir

Si el administrador elige esta opción se le mostrará en pantalla un mensaje de si desea realizar alguna otra operación. En el caso de elegir que sí, el programa se reiniciaría volviendo a pedir identificación al usuario, y en caso de elegir no, el programa guardaría los datos en disco y se cerraría.

1.5.2. Perfil profesor

Al usuario que ha sido identificado como profesor en el sistema se le mostrará una lista de todos los grupos y asignaturas que tiene asignado para el día actual (este sistema solo estará disponible para los días comprendidos entre lunes y viernes). El profesor tendrá que seleccionar el grupo con el que quiere trabajar en el sistema.

Una vez seleccionado el grupo, al profesor se le muestra en pantalla el siguiente menú:

- 1.- Listar alumnos.
- 2.- Cambiar de grupo.
- 3.- Salir.

1.- Listar alumnos

Cuando el profesor elija esta opción, le aparecerá una lista de todos los alumnos matriculados en ese grupo en concreto con la asignatura en concreto que ha seleccionado el profesor.

El sistema pedirá que introduzca el código de identificación de un alumno en concreto. Cuando esté seleccionado un alumno, al profesor le aparecerá en pantalla el siguiente menú:

- 1.- Ficha del alumno.
- 2.- Calificaciones del alumno.
- 3.- Faltas de asistencia.
- 4.- Volver.

1. **Ficha del alumno:** Mostrará todos los datos personales del alumno seleccionado (nombre, dirección y localidad) y preguntará al profesor si quiere modificar algún dato del alumno, en caso afirmativo preguntará que dato modificar y pasará a modificarlo introduciendo por teclado el nuevo valor.
2. **Calificaciones del alumno:** Mostrará todas las calificaciones del alumno en ese grupo y en esa asignatura. Y mostrará al profesor el siguiente menú por pantalla:
 - 1.- Modificar una calificación.
 - 2.- Borrar una calificación.
 - 3.- Añadir una calificación.
 - 4.- Volver.
 - a) **Modificar una calificación:** Permitirá seleccionar una calificación en concreto de la lista (usando la descripción) y nos preguntará la nueva calificación que será modificada.
 - b) **Borrar una calificación:** Permitirá seleccionar una calificación de la lista (usando la descripción) esa calificación será borrada del sistema.
 - c) **Añadir una calificación:** Permitirá que el profesor pueda añadir una nueva calificación al alumno seleccionado. Los datos que pedirá serán la fecha, la descripción de la calificación y la nueva nota.
 - d) **Volver:** Esta opción devuelve al profesor al menú de listar alumnos volviendo a mostrar los alumnos del grupo y asignatura que se activó en el momento.
3. **Faltas de asistencia:** mostrará por pantalla todas las faltas de asistencia del alumno en esa materia junto con el siguiente menú:
 - 1.- Modificar una falta de asistencia.
 - 2.- Borrar una falta de asistencia.
 - 3.- Añadir una falta de asistencia.
 - 4.- Volver.
 - a) **Modificar una falta de asistencia:** Permitirá al profesor seleccionar una falta de asistencia del alumno seleccionado, la fecha y la hora. Tras esto, permitirá cambiar la descripción de la falta y el estado de la misma.
 - b) **Borrar una falta de asistencia:** Permitirá al profesor seleccionar una falta de asistencia del alumno seleccionado, la fecha y la hora. Tras esto el sistema eliminará la falta del sistema.

- c) **Añadir una falta de asistencia:** permitirá al profesor añadir una falta de asistencia del alumno pidiendo que introduzca al sistema la fecha, hora, descripción de la falta, estado de la falta (injustificada, justificada o retraso). tras esto, el sistema añadirá la nueva falta y volverá a mostrar todas las faltas del alumno en la materia seleccionada.
 - d) **Volver:** Esta opción devuelve al profesor al menú de listar alumnos volviendo a mostrar los alumnos del grupo y la asignatura que se activó en el momento.
4. **Volver:** Devolverá al profesor al menú inicial donde se le volverá a mostrar la lista de todos los grupos y asignaturas que tiene asignado para el día actual.

2.- Cambiar de grupo

Volverá a mostrar todos los grupos que tiene ese profesor en el día de hoy y podrá volver a realizar todas las operaciones descritas anteriormente en el sistema.

3.- Salir

Si el profesor elige esta opción se le mostrará un mensaje de si desea realizar alguna otra operación. En el caso de elegir que sí, el programa se reiniciará volviendo a pedir identificación al usuario, y en caso de elegir no, el programa guardaría los datos en disco y se cerraría.

Capítulo 2

Pruebas

2.1. Introducción

Para realizar las pruebas usaremos utilizaremos la función `alta_Usuarios` del módulo `administrador.c` por lo que este capítulo estará centrado solo en dicha función:

```
1 //Cabecera: static void alta_Usuarios(usuarios **v_usuarios);
2 //Precondicion: v_usuarios debe ser un vector previamente inicializado.
3 //Postcondicion: crea un nuevo usuario en el sistema, es decir, anade una
4 //posicion mas al vector v_usuarios.
5
6 static void alta_Usuarios(usuarios **v_usuarios){
7     int j,i,condicion,condicion2;
8     char id_user[4];
9
10    condicion=1;
11    condicion2=0;
12
13    while(condicion!=0){
14        system("cls");
15        printf("Introduzca_un_identificador_de_usuario_disponible:_");
16        fflush(stdin);
17        scanf("%3s",id_user);
18        condicion2=0;
19        for(i=0;i<n_act_usuarios;i++){ //Controlamos la introduccion en el
20            //sistema de identificadores de usuarios ya existentes.
21            if(strcmp((*v_usuarios)[i].id_usuario,id_user)==0){
22                printf("ERROR:_Usuario_ ya _existente \n");
23                system("pause");
24                condicion2=1;
25            }
26        }
27        if(condicion2==0)
28            condicion=0;
29    }
30 }
```

```

31     printf("Identificador_disponible.\n");
32     system("pause");
33
34     *v_usuarios=(usuarios *)realloc((*v_usuarios),((n_act_usuarios)+1)*
35     sizeof(usuarios)); //Le anadimos el +1 puesto que estamos reservando
36     //memoria dinamicamente para un nuevo usuario, teniendo en cuenta que
37     //existe ya un vector reservado dinamicamente previamente.
38
39     system("cls");
40
41     //Dar de alta identificador de usuario.
42     strcpy((*v_usuarios)[n_act_usuarios].id_usuario,id_user);
43     system("cls");
44     //Dar de alta nombre de usuario.
45     printf("Por_favor ,_introduzca_el_nombre_del_usuario:_\n"
46     "(20_caracteres_como_maximo)\n");
47     fflush(stdin);
48     gets((*v_usuarios)[n_act_usuarios].Nomb_usuario);
49
50     system("cls");
51     //Dar de alta tipo de perfil de usuario.
52     printf("Elija_el_perfil_de_usuario_que_esta_creando.Seleccione_la_opcion"
53     "deseada:\n1.-Administrador.\n2.-Profesor.\nSeleccionar:_");
54     scanf("%d",&j);
55     if(j==1){
56         strcpy((*v_usuarios)[n_act_usuarios].Perfil_usuario,"Administrador");}
57     if(j==2){
58         strcpy((*v_usuarios)[n_act_usuarios].Perfil_usuario,"Profesor");}
59
60     system("cls");
61     //Dar de alta nombre de usuario.
62     printf("Por_favor ,_introduzca_el_nombre_de_usuario_con_el_que_se_"
63     "identificara_en_el_sistema:_(Tiene_un_maximo_de_5_caracteres.)\n");
64     fflush(stdin);
65     gets((*v_usuarios)[n_act_usuarios].Usuario);
66
67     system("cls");
68     //Dar de alta contraseña de usuario.
69     printf("Finalmente ,_introduzca_la_contraseña_del_usuario:_\n"
70     "(Tiene_un_maximo_de_8_caracteres_y/o_digitos)\n");
71     fflush(stdin);
72     gets((*v_usuarios)[n_act_usuarios].Contraseña);
73
74     puts("Usuario_anadido_correctamente.");
75     Sleep(2000);
76     system("cls");
77
78     (n_act_usuarios)++; //Al haber un usuario mas en el sistema ,
79                        //se incrementa la variable global.
80 }

```

2.2. Pruebas de caja blanca

Las pruebas de caja blanca se centran en cubrir el control de flujo del programa. Estas pruebas aseguran que la integración de todos los componentes del programa es correcta, por lo que se centra en la implementación eligiendo los casos de prueba que garanticen todos los posibles caminos de ejecución que puedan trazarse en el programa.

Para realizar la prueba de caja blanca, se utiliza la técnica de la ruta básica, la cual nos dice cuáles son los posibles caminos de ejecución que tiene el programa al que se le ha realizado dicha prueba.

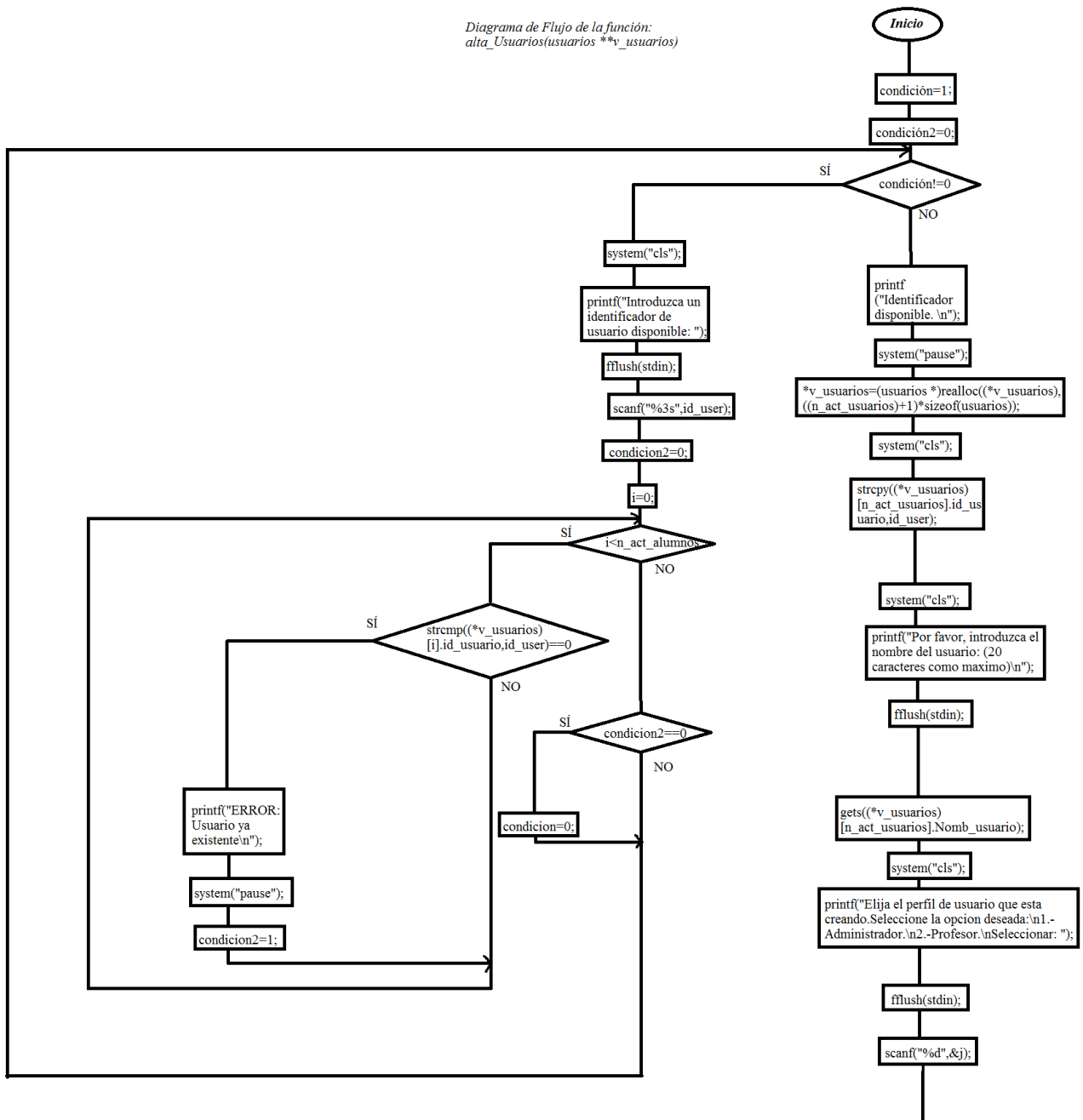
2.2.1. Prueba de la ruta básica

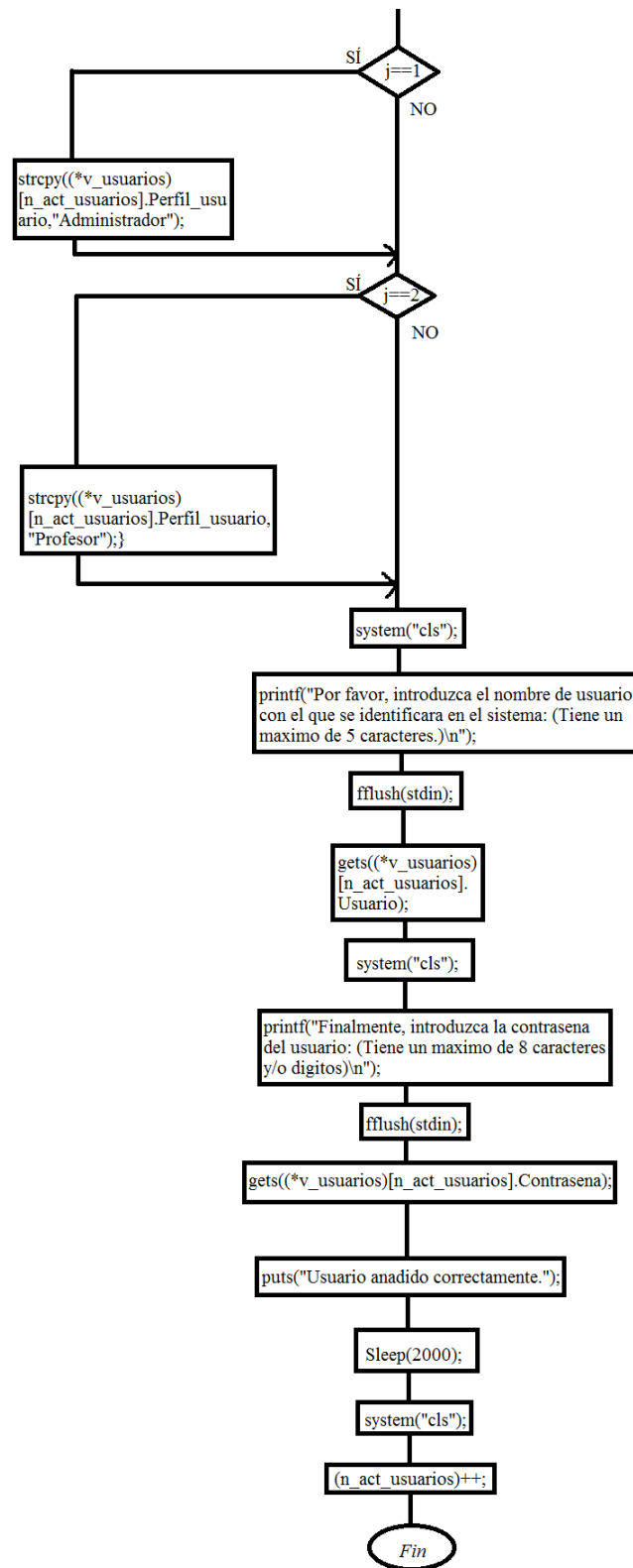
Para realizar esta prueba y con ello la prueba de caja blanca, hemos seleccionado la función `alta_Usuarios` del módulo `administrador.c` ya que tiene sentencias condicionales, de repetición y de asignación.

Para ello hemos hecho el diagrama de control de flujo de dicha función y hemos calculado su complejidad ciclomática, que nos dará el número de rutas posibles en dicha función.

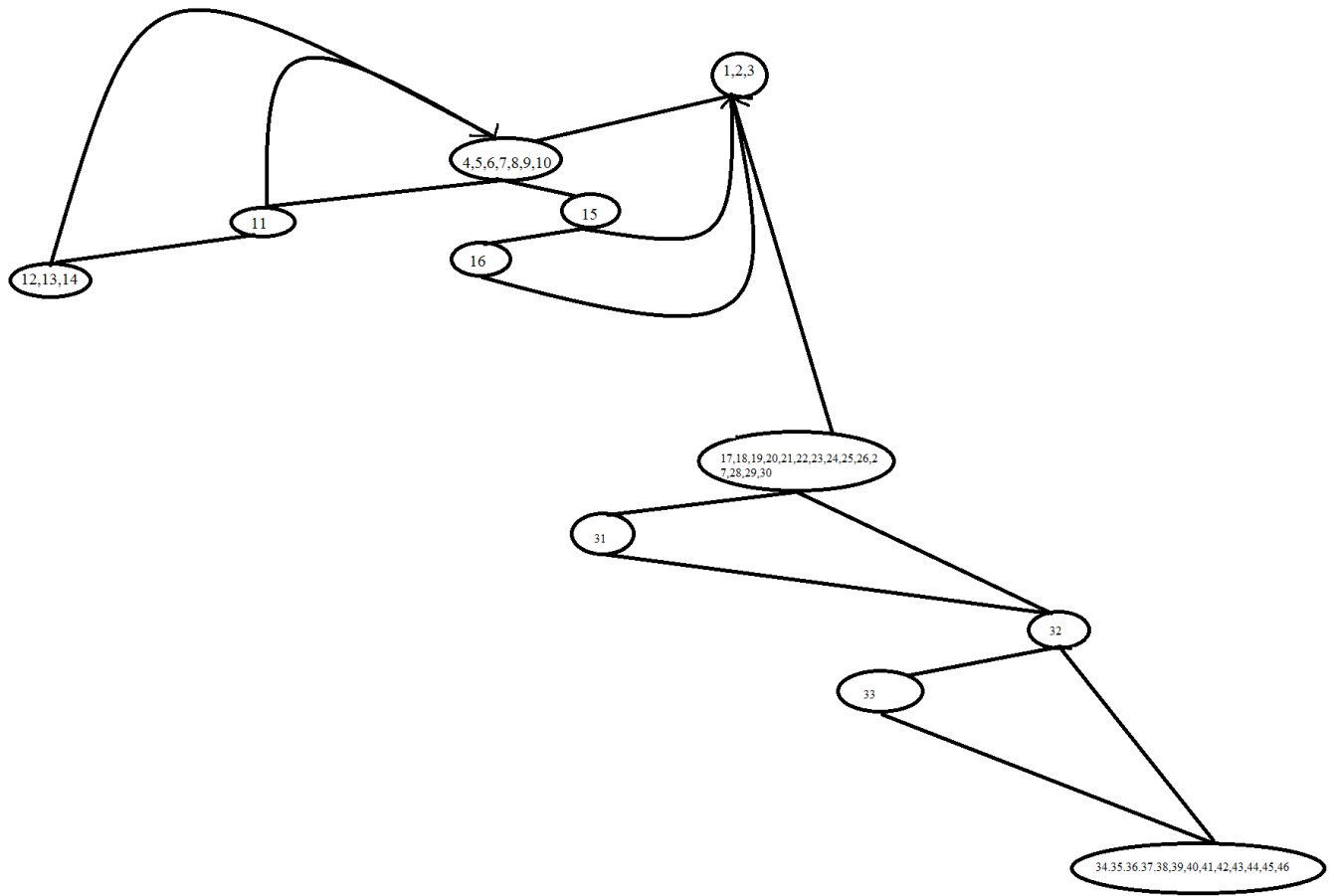
Diagrama de control de flujo

Diagrama de Flujo de la función:
alta_usuarios(usuarios **v_usuarios)





Complejidad ciclomática



La complejidad ciclomática se calcula de dos formas posibles:

- $NA - NN + 2$
- $NNP + 1$

Siendo: NA el número de aristas, NN número de nodos y NNP número de nodos predicado (nodos que contienen alguna condición).

Para esta función tenemos:

- $NA - NN + 2 \Rightarrow 16 - 11 + 2 = 7$
- $NNP + 1 \Rightarrow 6 + 1 = 7$

Por lo tanto, tenemos 7 posibles rutas que señalaremos con los números que indican los nodos de la complejidad ciclomática:

1. 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.

2. 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.
3. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.
4. 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.
5. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 16, 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.
6. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 4, 5, 6, 7, 8, 9, 10, 15, 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.
7. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1, 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46.

2.3. Pruebas de caja negra

Las pruebas de caja negra se realizan para determinar que cada función realiza la tarea para la que ha sido creada. Por lo tanto, se analiza la especificación de las funciones, su entrada y su salida.

En la función que estamos tratando (`alta_Usuarios`), podemos ver que en la precondition nos pide que el vector `v_usuarios` tiene que estar inicializado antes de realizar la llamada a esta función.

En su postcondición se establece su funcionalidad: creará un nuevo usuario en el sistema añadiendo una nueva posición al vector `v_usuarios`.

Diferentes casos de prueba a tener en cuenta:

- Si en la sentencia `if` de la línea 21 de código se cumple la condición, el programa habrá encontrado un usuario que tiene el mismo identificador que el que queremos introducir, por lo tanto, abortará la tarea de dar de alta un nuevo usuario y volverá a pedir un nuevo identificador. Si esta sentencia no se cumple, el programa continuará con el alta del usuario.
- El bucle de la línea 19 de código sirve para recorrer el vector de posiciones buscando un usuario que tenga el mismo identificador que el que hemos introducido por teclado. Cuando llegue al valor de la variable `n_act_usuarios` terminará el recorrido continuando con la siguiente sentencia condicional `if` de la línea 27.
- La sentencia condicional `if` de la línea 27 de código establece la condición de haber encontrado un usuario con el mismo identificador y establecerá a 0 la variable `condicion` en caso de no haber encontrado dicho usuario.
- El bucle más externo se encuentra en la línea 13 de código que es el que se encarga de controlar toda la gestión de la introducción del usuario de un identificador válido e inexistente en el sistema. Si no se introduce un identificador válido, lo seguirá exigiendo hasta que se le introduzca.
- Por último en las líneas 55 y 57 se encuentran dos sentencias condicionales que se encargan de fijar el rol que representarán los usuarios en el sistema, que será seleccionado en el registro de cada uno de ellos. Si se elige la opción “1” será un administrador y si se elige la opción “2” será un profesor.