DOCUMENTACIÓN INTERNA DEL PROYECTO

Nombre: José Joaquín Carral Fernández

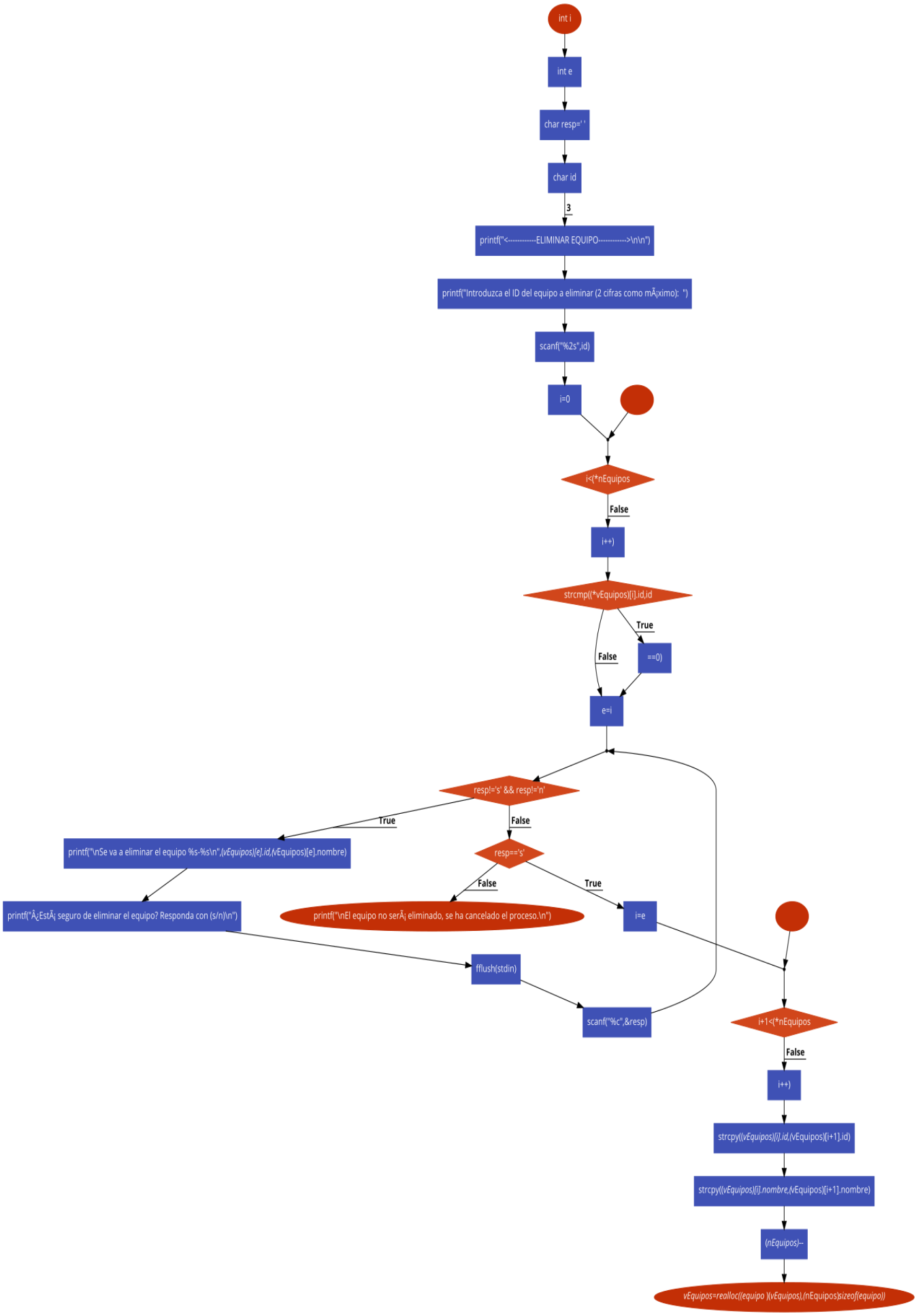
He seleccionado una función algo compleja de nuestros módulos, concretamente he cogido el siguiente código en c:

```
void eliminarEquipo(equipo **vEquipos,int *nEquipos){
    int i;
    int e;
    char resp=' ';
    char id[3]
   printf("<------------ELIMINAR EQUIPO------------>\n\n");
   printf("Introduzca el ID del equipo a eliminar (2 cifras como mÃ¡ximo):  ");
   scanf("%2s",id)
   for(i=0;i<(*nEquipos);i++){
       if(strcmp((*vEquipos)[i].id,id)==0){
          e=i;
       }
    }
  while(resp!='s' && resp!='n'){
       printf("\nSe va a eliminar el equipo %s-%s\n",(*vEquipos)[e].id,(*vEquipos)[e].nombre);
       printf("Â¿EstÃ¡ seguro de eliminar el equipo? Responda con (s/n)\n");
       fflush(stdin);
       scanf("%c",&resp);
    }
if(resp=='s'){
       for(i=e;i+1<(*nEquipos);i++){
          strcpy((*vEquipos)[i].id,(*vEquipos)[i+1].id);
          strcpy((*vEquipos)[i].nombre,(*vEquipos)[i+1].nombre);
       }
     (*nEquipos)--;
     *vEquipos=realloc((equipo *)(*vEquipos),(*nEquipos)*sizeof(equipo));
```

```
 }
else{
      printf("\nEl equipo no serÃ¡ eliminado, se ha cancelado el proceso.\n");
   }
}
```

Pruebas de caja blanca

Para realizar las pruebas de caja blanca lo primero que deberíamos hacer sería hacer un diagrama de flujo de la función la cual queremos hacer y sería (Se pondrá en la siguiente página).


int i
int e
char resp=' '
char id
3
printf("<----------ELIMINAR EQUIPO---------->\n\n")
printf("Introduzca el ID del equipo a eliminar (2 cifras como mÃ¡ximo): ")
scanf("%2s",id)
i=0
i<(*nEquipos
False
i++)
strcmp((*vEquipos)[i].id,id
True
False
==0)
e=i
resp!='s' && resp!='n'
True
False
printf("\nSe va a eliminar el equipo %s-%s\n",(vEquipos)[e].id,(vEquipos)[e].nombre)
resp=='s'
False
True
printf("Â¿EstÃ¡ seguro de eliminar el equipo? Responda con (s/n)\n")
printf("\nEl equipo no serÃ¡ eliminado, se ha cancelado el proceso.\n")
i=e
fflush(stdin)
scanf("%c",&resp)
i+1<(*nEquipos
False
i++)
strcpy((vEquipos)[i].id,(vEquipos)[i+1].id)
strcpy((vEquipos)[i].nombre,(vEquipos)[i+1].nombre)
(nEquipos)--
vEquipos=realloc((equipo)(vEquipos),(nEquipos)sizeof(equipo))

Caminos básicos:

- Camino 1: Empezaríamos con la inicializaciones y seguidamente pasamos a que si i<(*nEquipos) es **falso** incrementamos i, seguidamente si strcmp((*Equipos)[i].id,id es **falso** e=1. Después de esto tenemos una condición resp!='s' && resp!='n' que si es **falsa** la resp==s, seguidamente si resp==s es **falsa** se ejecuta el printf y no se elimina ningún equipo.

- Camino 2: Empezaríamos con las inicializaciones luego i<(*nEquipos) sería **falsa**, seguidamente strcmp((*Equipos)[i].id,id es **verdadera** por tanto va a ser ==0 y luego e=i, después de esto pasamos al bucle donde tiene la condición condición resp!='s' && resp!='n' que si la tomamos como **verdadera** se ejecutaría pero volvería de nuevo al bucle de la condición anterior y se repetiría infinitas veces la condición verdadera hasta que no digamos lo contrario.

- Camino 3: Empezaríamos con las inicializaciones luego i<(*nEquipos) sería **falsa,** luego strcmp((*Equipos)[i].id,id lo tomamos como **verdadero** y la comparación sería ==0 y la e=i, después de esto pasamos al bucle donde tiene la condición condición resp!='s' && resp!='n' que si la tomamos como **falsa** la resp==s, luego si esta la tomamos como **verdadera** i=e y se llevara acabo la eliminación del equipo mediante un bucle con la siguiente condición i+1<(*nEquipos) que tiene que ser **falsa**.

- Camino 4: Igual que el camino 3 solo que strcmp((*Equipos)[i].id,id es **falso** y solamente e=i lo demás todo igual que en el camino 3.

- Camino 5: Igual que el camino 2 solamente cambiando que strcmp((*Equipos)[i].id,id es **falsa** y a partir de ahí igual que en el camino 2.

- Camino 6: Igual que el camino 1 solo que poniendo strcmp((*Equipos)[i].id,id es **verdadero**.

A continuación, calcularemos la complejidad ciclomática de nuestra función:

V(G)=NNP+1=5+1=6 **V(g)=6**

Prueba de caja negra

Para realizar las pruebas de caja negra necesitamos saber la especificación de la función:

Cabecera: void eliminarEquipo(equipo **vEquipos,int *nEquipos)

Precondición: vEquipos y nEquipos los cuales están cargados y inicializados

Postcondición: Para finalizar el usuario, sí quiere y desea puede eliminar a un equipo.

Esta serie de pasos que acabamos de realizar se harían si el usuario estuviera con el deseo de eliminar a un equipo y para ello debería introducir su id y el nombre del equipo que quiere eliminar y seguidamente indicará que si quiere eliminarlo.