
INTRODUCCIÓN MEMORIA

Los conocimientos aplicados en esta práctica no corresponden con los aprendidos en la práctica, debido a problemas de tiempo, mala redacción del enunciado (inconcluso). He invertido mucho esfuerzo y a pesar de haber aprendido mucho en las clases y en el libro no he podido plasmar mi conocimiento. He intentado respetar las partes de un programa que me has enseñado en clase, la limpieza, el orden…
Pero todo esto, como digo, me ha resultado imposible, con tan poco tiempo, debido a la coincidencia de otras prácticas de otras asignaturas y la tardanza en dar conocimientos necesarios para la realización de la misma, es incompatible.

DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA

He dividido esta práctica en múltiples módulos:

-Módulo Principal: Este módulo es “Gesflota.cpp”. Un archivo ejecutable sobre el cual se desenvuelve todo el programa. Se compone de un switch en el que según la opción que seleccione el usuario le va a introducir a uno de los métodos que componen el menú. Para que el módulo principal tenga acceso a cada uno de los módulos secundarios he tenido que insertarlos como si fuesen “librerías” mediante “#include “NombreMódulo.h”. Por último, cabe destacar el uso del do-while para que no deje de ejecutarse el módulo principal y así, el programa entero, hasta que el usuario no seleccione la letra “S”.

-Módulos Secundarios: Estos son todos los métodos que componen el programa principal. Cada uno de ellos está desarrollado en su cabecera correspondiente.
Hay un módulo especial secundario que he llamado “BaseDeDatos”. Este se invoca en todos los demás módulos secundarios ya que es como el almacén de todos los datos recogidos a lo largo del programa. Las demás cabeceras son:

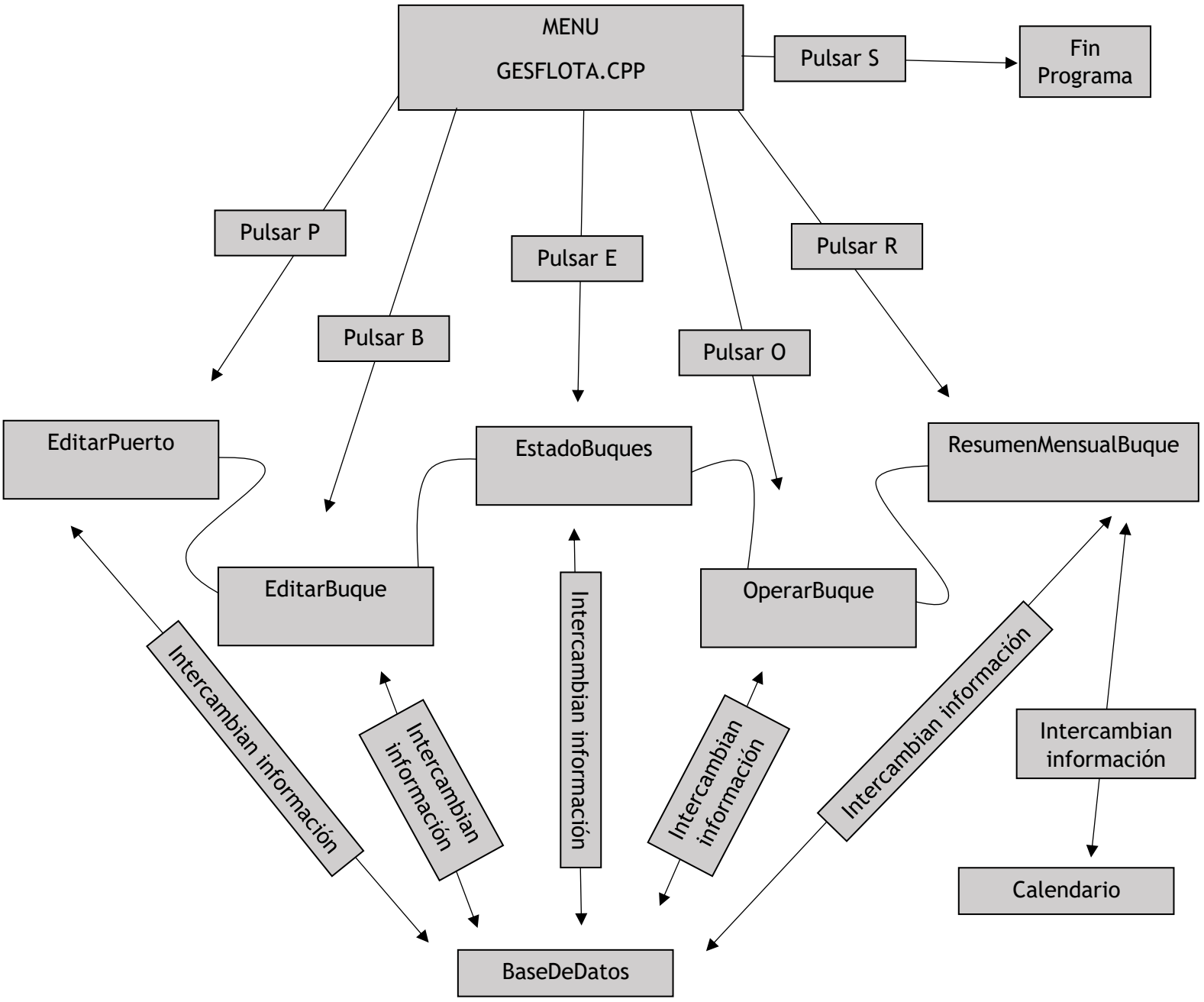
- EditarBuque
- OperarBuque
- EstadoBuques
- ResumenMensualBuques
- EditarPuerto
- Calendario (usado para ResumenMensualBuques)

Para entenderlo un poco mejor voy a hacer uso de una metáfora. Mi programa es como un cuerpo humano:

Cerebro: Módulo Principal “Gesflota”
Corazón: “BaseDeDatos” del Módulo Secundario
Extremidades y demás: Módulo Secundario.

En la siguiente hoja se verá un diagrama de todos los módulos y la interacción que tienen entre ellos.

DIAGRAMA DE LOS MÓDULOS



¿Por qué he elegido este sistema? Es más fácil ir menú por menú programando todas las necesidades requeridas, ya que luego la información creada en estos subprogramas luego se utiliza en los otros.

Con esta introducción explicaré todos los métodos dentro de los subprogramas.

EXPLICACIÓN MÉTODOS Y DATOS CABECERAS

"BaseDeDatos.h"

En este subprograma tenemos diferentes tipos predefinidos. Necesitamos crear dos tipos: Buques y Puertos.
Estos tipos son structs(registros) donde almacenamos información acerca de estos dos elementos. En el enunciado pide que tiene que haber máximo 10 puertos y 5 buques, como programador he creado puertos y buques por defectos que el usuario puede cambiar en la ejecución del programa a su gusto.

La información que guarda cada struct:

-Puertos= su identificador, su nombre y el tipo de puerto.
-Buques= su identificador, su nombre, la fecha de alta en el sistema y luego otros datos que se adquieren en otros métodos necesariamente como el tipo de carga que llevan, dónde se encuentra el buque…

```
typedef struct Buques{
  char identificador;
  typedef char Nombres[20];
  Nombres nombre;
  int dia;
  int mes;
  int anno;
  Nombres PuertoSalida;
  Nombres Carga;
  Nombres tipopuerto;
  Nombres PuertoOrigen;
  Nombres CargaOrigen;
  int identificadorPuerto;
};
Buques Buque1 = {'A',"Maine",25,7,2020,"HongKong","Vacio"};
Buques Buque2 = {'B',"Calypso",14,7,2021,"Amberes","Gasoil"};
Buques Buque3 = {'C',"Peel",3,3,2022,"HongKong","Crudo"};
Buques Buque4 = {'D',"Audaz",9,12,2022,"Valencia","Gasolina"};
Buques Buque5 = {'E',"Neptuno",12,11,2022,"Cagliari","Vacio"};
```

Luego tenemos otros métodos Carga1, Carga2, Carga3, Carga 4 y Carga 5. Estos métodos sirven para imprimir y saber si el buque lleva crudo o producto refinado. Se usa la función "strcmp" dentro del tipo predefinido string.

```
void Carga1(){

  typedef char Nombres[20];
  Nombres Crudo="Crudo";
  Nombres Fuel="Fuel";
  Nombres Gasoil="Gasoil";
  Nombres Gasolina="Gasolina";

  if(strcmp(Buque1.tipopuerto,"Yacimiento")==0){
    printf("%s",Crudo);}
  else{
    printf("%s,%s o %s",Fuel,Gasoil,Gasolina);}}
```

Por último, tenemos el método TipoPuerto. Este método sirve para saber en qué tipo de puerto están cada uno de los buques. Lo que hace es comparar si son iguales el Puerto de Origen del buque y los diferentes puertos existentes. Si coinciden, copian el tipo de Puerto en el tipo de puerto de los buques.

```
if(strcmp(Buquel.PuertoSalida,Puerto1.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto1.tipopuerto);
  Buquel.identificadorPuerto=Puerto1.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto2.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto2.tipopuerto);
  Buquel.identificadorPuerto=Puerto2.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto3.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto3.tipopuerto);
  Buquel.identificadorPuerto=Puerto3.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto4.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto4.tipopuerto);
  Buquel.identificadorPuerto=Puerto4.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto5.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto5.tipopuerto);
  Buquel.identificadorPuerto=Puerto5.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto6.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto6.tipopuerto);
  Buquel.identificadorPuerto=Puerto6.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto7.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto7.tipopuerto);
  Buquel.identificadorPuerto=Puerto7.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto8.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto8.tipopuerto);
  Buquel.identificadorPuerto=Puerto8.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto9.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto9.tipopuerto);
  Buquel.identificadorPuerto=Puerto9.identificador;}
if(strcmp(Buquel.PuertoSalida,Puerto10.nombre)==0){
  strcpy(Buquel.tipopuerto,Puerto10.tipopuerto);
  Buquel.identificadorPuerto=Puerto10.identificador;}
```

Tanto Carga() como TipoPuerto() son utilizados en OperarBuque.h que veremos más adelante.

“EditarPuerto.h”

En este menú se solicita información para editar o crear un puerto, dependiendo del identificador que el usuario seleccione. Luego lo que cambia es el nombre y el tipo de puerto que el usuario quiera.

Para ello se utilizan diferentes variables según el tipo de dato que queremos manipular:

```
void EditarPuerto(){

  int identificador;
  typedef char Nombres[20];
  Nombres nombre;
  char tipoPuerto;
  char SN;
```

Lo demás del programa se basa en la recogida de datos mediante scanf y el while para que los datos introducidos sean válidos

```
12  printf("\n\n\n\n\nEditar Puerto:\n");
13  printf("\n");
14  printf("       Identificador (numero entre 1 y 10)?");
15  scanf("%d", &identificador);
16  while(identificador<1 || identificador>10){
17    printf("       Identificador (numero entre 1 y 10)?");
18    scanf("%d", &identificador);
19  }
20  printf("       Nombre (entre 1 y 20 caracteres)?");
21  scanf("%s", &nombre);
22  while(strlen(nombre)>20 || strlen(nombre)<1){
23  printf("       Nombre (entre 1 y 20 caracteres)?");
24  scanf("%s", &nombre);
25  }
26  do{
27  printf("       Tipo(Y-Yacimiento, R-Refineria, D-Deposito)?");
28  scanf(" %c",&tipoPuerto);
29  }
30  while(tipoPuerto!='Y' && tipoPuerto!='R' && tipoPuerto!='D');
31  printf("\n");
32  printf("IMPORTANTE: Esta opcion borra los datos anteriores.\n");
33  printf("Son correctos los nuevos datos (S/N)?");
34  scanf(" %c", &SN);
35  while(SN!='S'&&SN!='N'){
36  printf("Son correctos los nuevos datos (S/N)?");
37  scanf(" %c", &SN);
38  }
```

Por último, si se quieren confirmar los cambios, es entonces, donde se tienen que hacer los cambios en nuestra base de datos creada llamada “BaseDeDatos.h”. Para eso utilizamos un switch con 10 casos que son los 10 identificadores de los 10

posibles puertos existentes, que ya tenían una información correspondiente dada por mí.

```
switch(identificador){

   case 1:
   strcpy(Puerto1.nombre,nombre);
   switch(tipoPuerto){
   case 'Y':
   strcpy(Puerto1.tipopuerto,"Yacimiento");break;
   case 'R':
   strcpy(Puerto1.tipopuerto,"Refineria");break;
   case 'D':
   strcpy(Puerto1.tipopuerto,"Deposito");break;
   }
   break;
```

Los otros casos, tienen la misma programación, pero cambiando Puerto1 por Puerto2. Este formato de subprograma se repite en el siguiente subprograma “EditarBuque.h” por lo tanto, no nos pararemos demasiado.

“EditarBuque.h”

Como he dicho, tiene el mismo funcionamiento que el anterior subprograma, por lo tanto, solo voy a destacar lo nuevo.

Se solicita el puerto de origen, y según el identificador, en el switch se realizarán los cambios necesarios en la “BaseDeDatos.h”

```
printf("       Puertos posibles para la ubicacion inicial del buque:\n");
printf("          %d - %s\t\t\tTipo: %s\n",Puerto1.identificador,Puerto1.nombre,Puerto1.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto2.identificador,Puerto2.nombre,Puerto2.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto3.identificador,Puerto3.nombre,Puerto3.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto4.identificador,Puerto4.nombre,Puerto4.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto5.identificador,Puerto5.nombre,Puerto5.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto6.identificador,Puerto6.nombre,Puerto6.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto7.identificador,Puerto7.nombre,Puerto7.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto8.identificador,Puerto8.nombre,Puerto8.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto9.identificador,Puerto9.nombre,Puerto9.tipopuerto);
printf("          %d - %s\t\t\tTipo: %s\n",Puerto10.identificador,Puerto10.nombre,Puerto10.tipopuerto);
printf("       Identificador de puerto inicio?");
scanf("%d", &identificadorPuertoInicio);
while(identificadorPuertoInicio>10 || identificadorPuertoInicio<1){
scanf("%d", &identificadorPuertoInicio);}
```

Aquí un caso de los 5 buques con los cambios que tiene que hacer según la información que edita el usuario:

```
switch(identificadorPuertoInicio){
  case 1:
  strcpy(Buque1.PuertoSalida,Puerto1.nombre);
  break;
  case 2:
  strcpy(Buque1.PuertoSalida,Puerto2.nombre);
  break;
  case 3:
  strcpy(Buque1.PuertoSalida,Puerto3.nombre);
  break;
  case 4:
  strcpy(Buque1.PuertoSalida,Puerto4.nombre);
  break;
  case 5:
  strcpy(Buque1.PuertoSalida,Puerto5.nombre);
  break;
  case 6:
  strcpy(Buque1.PuertoSalida,Puerto6.nombre);
  break;
  case 7:
  strcpy(Buque1.PuertoSalida,Puerto7.nombre);
  break;
  case 8:
  strcpy(Buque1.PuertoSalida,Puerto8.nombre);
  break;
  case 9:
  strcpy(Buque1.PuertoSalida,Puerto9.nombre);
  break;
  case 10:
  strcpy(Buque1.PuertoSalida,Puerto10.nombre);
```

“EstadoBuques.h”

Este subprograma es distinto a los otros porque el usuario no tiene ninguna interacción, nada más que la lectura. Se recogen todos los datos actuales de ese momento (incluso si se hicieron ediciones) sobre los buques y los imprime en una forma de tabla. Simplemente hay que respetar los márgenes y los espacios en blanco de los printf y recoger datos de las anteriores cabeceras.

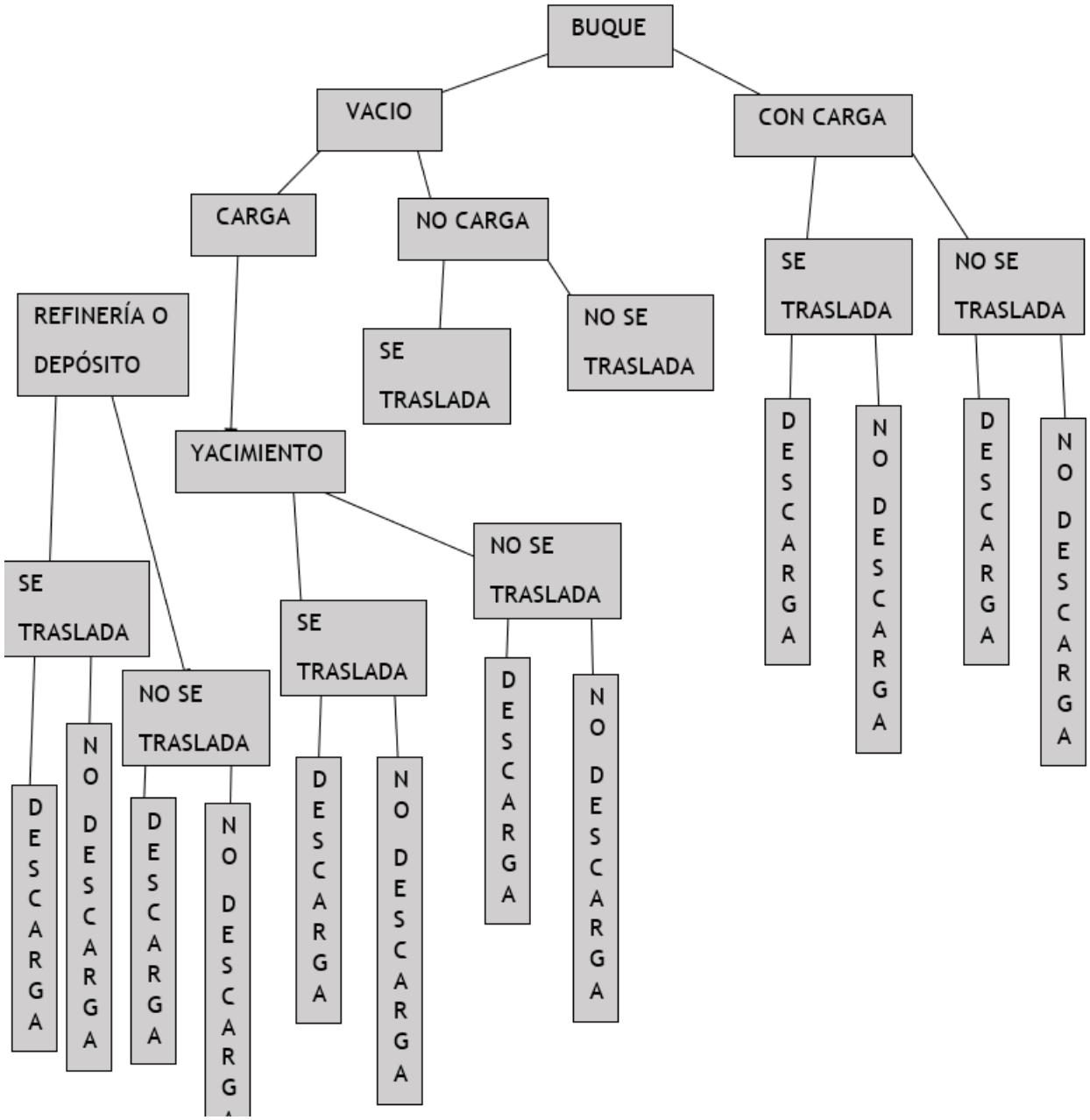
```
void EstadoBuques(){

printf("                    Estado Buques                    \n\n");
printf("Id    Nombre            Puerto            Ultima Fecha          Carga\n\n");
printf("%c     %s\t\t  %s\t\t%d/%d/%d\t     %s\n",Buque1.identificador,Buque1.nombre,Buque1.PuertoSalida,Buque1.dia,Buque1.mes,Buque1.anno,Buque1.Carga);
printf("%c     %s\t\t  %s\t\t%d/%d/%d\t     %s\n",Buque2.identificador,Buque2.nombre,Buque2.PuertoSalida,Buque2.dia,Buque2.mes,Buque2.anno,Buque2.Carga);
printf("%c     %s\t\t  %s\t\t%d/%d/%d\t     %s\n",Buque3.identificador,Buque3.nombre,Buque3.PuertoSalida,Buque3.dia,Buque3.mes,Buque3.anno,Buque3.Carga);
printf("%c     %s\t\t  %s\t\t%d/%d/%d\t     %s\n",Buque4.identificador,Buque4.nombre,Buque4.PuertoSalida,Buque4.dia,Buque4.mes,Buque4.anno,Buque4.Carga);
printf("%c     %s\t\t  %s\t\t%d/%d/%d\t     %s\n",Buque5.identificador,Buque5.nombre,Buque5.PuertoSalida,Buque5.dia,Buque5.mes,Buque5.anno,Buque5.Carga);

}
```

“OperarBuque.h”

Esta parte del programa es bastante inconclusa en el enunciado. Lo que permite es cargar un buque vacío, trasladar la carga y depositarla en otro puerto. Pero según el enunciado solo se realiza esto si el buque inicialmente está descargado. Lo cuál lo hace absurdo porque casi ningún buque va a estar vacío y si se carga un buque y no se descarga, ese buque ya no se puede operar a no ser que se modifique desde “EditarBuque.h”. Por lo tanto, lo que he añadido, es que este subprograma sea funcional y que los buques que ya tienen carga sean operativos. Esto agranda muchísimo el problema. He realizado un diagrama para entender bien este subprograma:



En total tenemos 16 casos y luego son 5 buques en total, por lo tanto, el programa se vuelve muy complejo. El programa hace estos pasos:

1. Comprobar si tiene carga el buque.
2. Si tiene la posibilidad de cargar el buque, quiere cargar o no cargar.
3. Quiere trasladarse hacia otro puerto o no.
4. Si lleva carga tener la posibilidad de descargar o no, depende del usuario.
5. Resumen de la operación.

1.

```
switch(identificador){
  case 'A': //BUQUE A//
if(strcmp(Buque1.Carga,"Vacio")==0){ //SI EL BUQUE ESTA VACIO//
```

2.

```
if(strcmp(Buque1.Carga,"Vacio")==0){ //SI EL BUQUE ESTA VACIO//
printf("       El buque %s esta vacio en %s\n",Buque1.nombre,Buque1.PuertoSalida);
strcpy(Buque1.PuertoOrigen,Buque1.PuertoSalida);
TipoPuerto(); //Detecta que tipo de carga se puede cargar segun el puerto en el que este//
printf("       Se puede cargar:");Carga1();
printf("\n       Quiere realizar la carga (S/N)?");
scanf(" %c", &SN);
while(SN!='S'&&SN!='N'){
printf("       Quiere realizar la carga (S/N)?");
scanf(" %c", &SN);
}
if(SN=='S'){ //el buque estando vacio se carga//
```

3.

```
if(SN=='S'){ //el buque estando vacio se carga//
  if(strcmp(Buque1.tipopuerto,"Yacimiento")!=0){//si esta en una refineria o deposito//
printf("       Se puede cargar:(1 = Fuel, 2 = Gasoil, 3 = Gasolina)");
scanf("%d",&carga);
while(carga>3 || carga<1){
scanf("%d",&carga);}
switch(carga){
  case 1:
  strcpy(Buque1.Carga,"Fuel");
  break;
  case 2:
  strcpy(Buque1.Carga,"Gasoil");
  break;
  case 3:
  strcpy(Buque1.Carga,"Gasolina");
  break;
}printf("       Duracion de la carga en dias?");
scanf("%d",&cargadias);
printf("\n");
printf("       Quiere realizar el traslado (S/N)?");
  scanf("%c", &SN);
  while(SN!='S'&&SN!='N'){
  scanf("%c", &SN);}
  if(SN=='S'){ //se traslada
  printf("       Puertos de posible destino del buque:\n");
  if(strcmp(Buque1.PuertoSalida,Puerto1.nombre)!=0){
```

```
  if(SN=='S'){ //se traslada
  printf("       Puertos de posible destino del buque:\n");
  if(strcmp(Buque1.PuertoSalida,Puerto1.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto1.identificador,Puerto1.nombre,Puerto1.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto2.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto2.identificador,Puerto2.nombre,Puerto2.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto3.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto3.identificador,Puerto3.nombre,Puerto3.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto4.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto4.identificador,Puerto4.nombre,Puerto4.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto5.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto5.identificador,Puerto5.nombre,Puerto5.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto6.nombre)!=0){
  printf("            %d - %s\t\t\t\tTipo:%s\n",Puerto6.identificador,Puerto6.nombre,Puerto6.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto7.nombre)!=0){
  printf("            %d - %s\t\t\t\tTipo:%s\n",Puerto7.identificador,Puerto7.nombre,Puerto7.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto8.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto8.identificador,Puerto8.nombre,Puerto8.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto9.nombre)!=0){
  printf("            %d - %s\t\t      Tipo:%s\n",Puerto9.identificador,Puerto9.nombre,Puerto9.tipopuerto);}
  if(strcmp(Buque1.PuertoSalida,Puerto10.nombre)!=0){
  printf("            %d - %s\t\t     Tipo:%s\n",Puerto10.identificador,Puerto10.nombre,Puerto10.tipopuerto);}
printf("       Identificador del puerto destino?");
scanf("%d", &identificadorP);
while(identificadorP>10 || identificadorP<1 || identificadorP==Buque1.identificadorPuerto){
  printf("       Identificador del puerto destino?");
  scanf("%d", &identificadorP);}
  printf("       Duracion del traslado en dias?");
scanf("%d",&trasladodias);
printf("\n");
```

4.

```
printf("       Quiere realizar la descarga (S/N)?");
scanf("%c", &SN);
while(SN!='S'&&SN!='N'){
scanf("%c", &SN);}
if(SN=='S' && strcmp(Buque1.tipopuerto,"Deposito")==0){ //buque de una refineria o deposito que se traslada
printf("       Duración de la descarga en días?");
  scanf("%d",&descargadias);
  printf("\n");
```

5.

```
     printf("              Resumen de la operacion:\n");
      printf("        Fecha comienzo: %d/%d/%d\n",dia,mes,anno);
      printf("        Puerto origen: %s\n",Buque1.PuertoOrigen);
      printf("        Tipo de carga: %s\n",Buque1.Carga);
      printf("        Duracion carga: %d\n",cargadias);
      printf("        Puerto destino: %s\n",Buque1.PuertoSalida);
      printf("        Duracion del traslado: %d\n",trasladodias);
      printf("        Duracion de la descarga: %d\n",descargadias);
//esta seguro de la operacion//
```

Habría que poner si está seguro de la operación y si no, reestablecer los datos de

origen, pero el programa entonces se hace muy extenso y no tiene ninguna "skill" diferente a la hora de realizar este programa. Me he dado cuenta de un problema también y es que al hacer la parte 4 así, a veces no se hacen traslados por lo tanto antes de poner el resumen de la operación igualar a 0 todas las variables que no se utilicen, porque sino sale en pantalla 4193…

"ResumenMensualBuque.h"

Esta parte de la práctica es la que tiene interacción con la parte 4. Se encarga de imprimir en forma de calendario el estado de los Buques. Es como la cabecera "EstadoBuques.h" pero datando la fecha de las operaciones de los buques. Esta parte de la práctica no me ha dado tiempo debido a lo que se me ha complicado la parte 4 debido a ese cambio. De todas formas, voy a narrar como se podría hacer:

Tendríamos que tener en cuenta mediante un switch los 5 casos de los 5 buques existentes. Después de eso, si coincide mes y año seleccionado, con alguna operación, imprimir en pantalla un calendario en el que se sustituye las fechas en las que corresponden los días de carga, traslado y descarga recogiéndolos del subprograma anterior. Para ello, en el calendario dentro de la carpeta tendrá un método que se llame imprimir calendario. Luego con bools si coincide T1/C/T2 imprimir en vez de números, las diferentes letras.

Por último poner el resumen por escrito de todo lo que se ve en el calendario y sería mediante printf e introducir las respectivas variables.

EJECUTANDO EL PROGRAMA

Para ejecutar el programa hay que abrir el archivo practica4.exe. El programa funciona perfectamente, salvo algún aspecto visual probablemente y la parte del calendario. Si se quiere leer/modificar el código habría que abrir gesflota.cpp y las demás cabeceras "NombreCabecera.h".