PRACTICA 4:

PRACTICA FINAL: TIPOS DE DATOS ESTRUCTURADOS.

Ismael García Hammad y Carlos Sánchez Mora.

22 de Enero de 2019.

Grupo Nº 13, Turno del Jueves.

Fecha de entrega: 23 de Enero de 2019.

ÍNDICE

Objetivos………………………………………..…………3

Diseño de las estructuras de datos…………….4

Diagrama de descomposición modular………5

Código fuente…………………………………………...6-

Pruebas de ejecución…………………………………

# *OBJETIVOS:*

En esta práctica desarrollaremos una aplicación completa organizada con ayuda de menús de opciones que se

mostrarán en pantalla de forma repetitiva al usuario. El diseño y la implementación de la aplicación debe

llevarse a cabo teniendo en cuenta los conceptos asociados a la programación modular y estructurada.

Trabajaremos con todos los conceptos teóricos estudiados en clase: estructuras de control, tipos de datos

simples y tipos de datos estructurados.

# *DISEÑO DE LAS ESTRUCTURAS DE DATOS:*

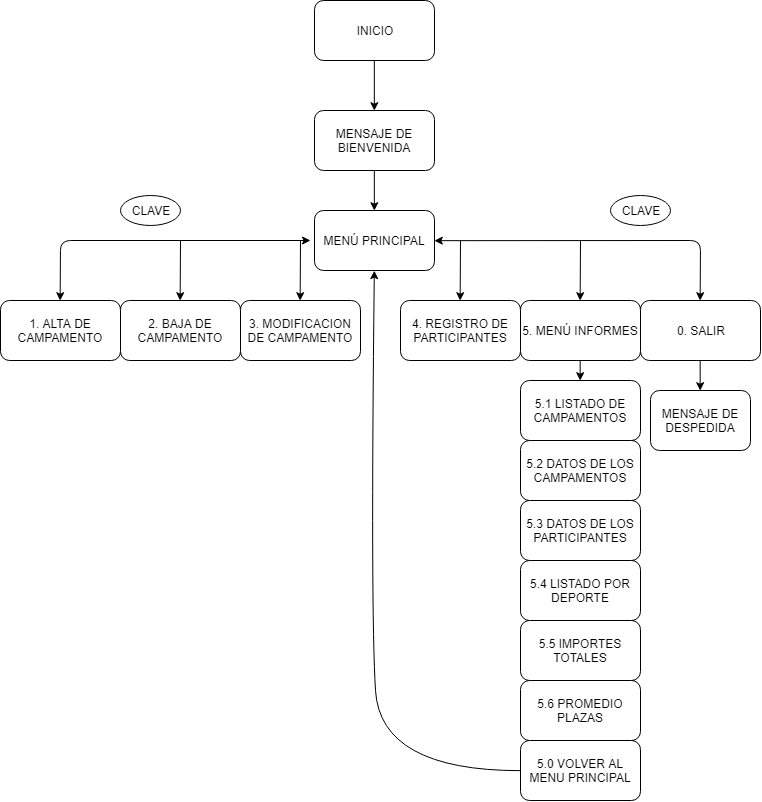
**POR CADA CAMPAMENTO:**

|  |  |
| --- | --- |
| **LUGAR**  (40 CARACTERES) | **MES**  (DESDE MAYO HASTA OCTUBRE) |
| **TURNO**  1 o 2 (PRIMERA O SEGUNDA QUINCENA DEL MES, RESPECTIVAMENTE) | **EDAD**  (ENTRE 14 Y 18 AÑOS) |
| **DEPORTE**  2 DEPORTES (20 CARACTERES) | **Nº DE PLAZAS**  ENTRE 5 Y 10 PLAZAS |

**POR CADA PARTICIPANTE:**

|  |  |
| --- | --- |
| **NOMBRE**  ( 40 CARACTERES) | **DNI**  ( 9 CARACTERES) |
| **EDAD ACTUAL**  DE 14 A 18 AÑOS | **SEXO**  HOMBRE O MUJER |

# *DIAGRAMA DE DESCOMPOSICIÓN MODULAR:*

**

# *CÓDIGO FUENTE:*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Alumno 1: Ismael García Hammad Alumno 2: Carlos Sánchez Mora

Grupo: 13 Fecha: 29 de Noviembre de 2018

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*Declaración de Librerias:\*\*/

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

#define MAX\_CLAVE 8

//Declaración de tipos de datos

typedef struct Plazas {

char Nombre[40];

int DNInum;

char DNIletra;

int Edad;

char Sexo;

}tRegisPlazas;

typedef struct Deportes {

char Deporte[20];

int Semana1, Semana2;

}tRegisDeportes;

typedef struct Campamento{

char clave[MAX\_CLAVE];

char Lugar [40];

char Mes[10];

int Plazas; //numero de plazas. Se inicializa a -1 si esta libre el campamento.

int ocupadas;

tRegisPlazas Asistentes[10];

int RangoEdadMin, RangoEdadMax;

int Precio;

int Semana1,Semana2;

tRegisDeportes Deportes[2];

};

/\*\*Declaración de Subprogramas:\*\*/

int MensajeBienvenida(void);

int MenuPrincipal();

void MensajeDespedida();

int MenuInformes();

void Pausa();

int BajaCamp(struct Campamento \*Campamento);

int RegistroParticipantes(struct Campamento \*Campamento);

void AltaCampamentos(struct Campamento \*Campamento);

void inicializar\_campamentos(struct Campamento \*Campamento);

int comprobar\_clave\_igual(struct Campamento \*Campamento,char \*Clave);

void ModificarCampamentos(struct Campamento \*Campamento);

int validar\_clave(char \*Clave);

/\*\* Declaracion de Subprogramas del menu informes:\*\*/

void ListadoCampamentos(struct Campamento \*Campamento);

void ConsultaCampamentos(struct Campamento \*Campamento);

void ConsultaDatosParticipante();

void ListadoDeporte();

void ImportesTotales();

void PromedioPlazas();

void Salir();

int Menu\_informes();

int main(){

char Clave;

int opcion;

struct Campamento campamentos[7];

inicializar\_campamentos(campamentos);

MensajeBienvenida();

do{

opcion = MenuPrincipal();

system("cls");

switch (opcion) {

case 1: AltaCampamentos( campamentos );

break;

case 2: BajaCamp(campamentos);

break;

case 3: ModificarCampamentos(campamentos);

break;

case 4: RegistroParticipantes(campamentos);

break;

case 5: MenuInformes();

break;

case 0: MensajeDespedida();

break;

default : printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR FAVOR, INGRESE UN NUMERO VALIDO.\n ");

}// FIN SWITCH

}while (opcion!=0);

return 0;

} // FIN MAIN

//---------------------------------------------------------------------------------------------------------------------------------------------

// ALGORITMOS DE LOS SUBPROGRAMAS:

int validar\_clave(char \*Clave)

{

int i=0;

short ClaveCorrecta;

while ((Clave[i]>='A')&&(Clave[i]<='Z')&&(i<4)){

i++; //aumenta el indice de uno en uno

}

if (i==4){

if (Clave[4]=='\_'){

i++;

while ((Clave[i]>='0')&&(Clave[i]<='9')&&(i<7)){

i++;

}

if(i==7)

ClaveCorrecta=1;

else

ClaveCorrecta=0;

}else

ClaveCorrecta=0;

}else

ClaveCorrecta=0;

return ClaveCorrecta;

}

int comprobar\_clave\_igual(struct Campamento \*Campamento, char \*Clave)

{

int i;

for (i=0;(i<7)&& (strcmp((Campamento+i)->clave,Clave) != 0);i++);

if (i<7)

{

printf("la clave introducida ya existe, por favor introduce una clave que no haya sido dada de alta \n");

Pausa();

return 1;

}

else

{

printf(" LA CLAVE ES CORRECTA \n");

Pausa();

return 0;

}

}

void inicializar\_campamentos(struct Campamento \*Campamento)

{

int i;

for(i=0;i<7;i++)

{

(Campamento+i)->Plazas=-1;

(Campamento+i)->ocupadas=0;

strcpy((Campamento+i)->clave," ");

}

}

int BajaCamp(struct Campamento \*Campamento)

{

char clave\_baja[8];

int i;

int opcion=0;

printf(" \n INTRODUCE LA CLAVE DEL CAMPAMENTO A DAR DE BAJA \n");

gets(clave\_baja);

for (i=0;(i<7)&& strcmp((Campamento+i)->clave,clave\_baja);i++); //algoritmo de busquedA DE ESA CLAVE para borrar sus datos asociados

if (i<7) // Existe esa clave?

{

if ((Campamento+i)->ocupadas==0)

{

printf(" \n Desea dar de baja el campamento: S=1/N=0 \n ");

scanf("%i",&opcion);

}

else // si las plazas no estan a 0, esque hay plazas asignadas, luego no se puede dar de baja el campamento.

{

printf(" \n imposible dar de baja el campamento, ya tiene plazas asignadas \n");

}

if (opcion=1)

{

(Campamento+i)->Plazas=-1; // se ponen las plazas a -1, como al principio

strcpy((Campamento+i)->clave," "); //7 espacios vacios a clave

printf(" \n USTED HA DADO DE BAJA EL CAMPAMENTO SELECCIONADO. \n");

return 0;

}

else

{

printf(" \n ha abortado dar de baja el campamento \n");

return 0;

}

}

else

{

printf(" no existe esa clave asociada a un campamento \n");

}

}

void ModificarCampamentos(struct Campamento \*Campamento)

{

char clave\_modificar[7];

int i;

printf("\n INTRODUCE una clave campamento para modificar \n");

gets(clave\_modificar);

for (i=0;(i<7)&& strcmp((Campamento+i)->clave,clave\_modificar);i++); //para tener acceso al struct de datos desde el 0-7, le ponemos un indice que recorra.

if (i<7)

{ //peticion (modificacion) de datos:

printf(" \n aqui puede modificar los datos: \n");

printf(" \n ---------------------------------------- \n");

printf("numero maximo de plazas habiles en el campamento: \n");

scanf("%i",&(Campamento+i)->Plazas);

printf("rango de edad minimo comprendido: \n");

scanf("%i",&(Campamento+i)->RangoEdadMin);

printf("rango de edad maximo comprendido: \n");

scanf("%i",&(Campamento+i)->RangoEdadMax);

printf("cual sera el precio del campamento: \n");

scanf("%i",&(Campamento+i)->Precio);

printf(" primera semana de duracion del campamennto: \n");

scanf("%i",&(Campamento+i)->Semana1);

printf(" segunda semana de duracion del campamennto: \n");

scanf("%i",&(Campamento+i)->Semana2);

printf(" que mes se realizara: \n ");

scanf("%s",&(Campamento+i)->Mes);

fflush(stdin);

printf(" donde tendra lugar el campamento: \n");

scanf("%s",&(Campamento+i)->Lugar);

}

else

{

printf(" La clave introducida no corresponde con ningun campamento. \n");

}

}

int MensajeBienvenida(void)

{

printf("\n\n\n \t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

printf("\n\n\n \t\t\t\t\t BIENVENIDO A CAMP ORGANIZER!\n");

printf("\n\n\n \t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

system (" \n\n\n\n\n \t\t\t\t\t\t pause \n\n\n\n\n ");

system(" \n\n\n\n\n \t\t\t\t\t\t\t cls");

}

int MenuPrincipal()

{

int opcion;

printf("\n\n \t\t\t\t MENU PRINCIPAL \n\n");

printf(" \t\t\t\t ---------------- \n\n");

printf(" \t\t\t\t 1.- Alta de Campamentos. \n\n");

printf(" \t\t\t\t 2.- Baja de Campamentos. \n\n");

printf(" \t\t\t\t 3.- Modificacion de Campamentos. \n\n");

printf(" \t\t\t\t 4.- Registro de Participantes. \n\n");

printf(" \t\t\t\t 5.- Informes. \n\n");

printf(" \t\t\t\t 0.- Salir. \n\n");

scanf("%i",&opcion);

fflush(stdin);

return (opcion); // devolvemos opcion al main

}

//\*\*\*\* subprogramas menu informes\*\*\*\*\*\*\*\*\*\*\*\*\*\*///////////////////////

void ListadoCampamentos(struct Campamento \*Campamento)

{

int i;

char Cadena[5];

system("cls");

printf("\n\n\n\t\t \*\*\*\*\*\*\*\*\*LISTADO GENERAL DE CAMPAMENTOS\*\*\*\*\*\*\*\*\*");

printf("\n\t\t -----------------------------------------------\n\n");

printf("\n\n\n\t\t CLAVE LUGAR MES SEMANAS PLAZAS RANGO DE EDAD PRECIO");

printf("\n\t\t ---------------------------------------------------------------------------------------------\n\n");

//algoritmo que diferencia las semanas del campamento "1 y 2" o "3 y 4".

for(i=0;i<7;i++)

{

if (((Campamento+i)->Semana1==1)&&((Campamento+i)->Semana2==2))

{

strcpy(Cadena, "1 y 2");

}

else

{

strcpy(Cadena, "3 y 4");

}

// algoritmo para diferenciar los campamentos dados de alta. Como las plazas estan inicializadas a -1, cuando estan a -1 significa que estan vacias, sino no.

if((Campamento+i)->Plazas!=-1)

{

printf("%s \t %s \t\t %s \t\t %i \t\t %i \t %i - %i \t\t %i \n",(Campamento+i)->clave,(Campamento+i)->Lugar,(Campamento+i)->Mes, Cadena, (Campamento+i)->Plazas, (Campamento+i)->RangoEdadMin, (Campamento+i)->RangoEdadMax, (Campamento+i)->Precio);

}

Pausa();

}// FIN for.

}

void ConsultaCampamentos(struct Campamento \*Campamento)

{

int i;

char Clave\_consulta[7];

printf("\n\t\t introduzca la clave del campamento que desea consultar: \n"); // pedimos al usuario que digite una clave para llevar a cabo la busqueda

gets(Clave\_consulta);

for (i=0;(i<7)&& strcmp((Campamento+i)->clave,Clave\_consulta);i++); // se comprueba la existencia de esa clave, comparandola .

if (i<7)

{

printf("Mostrando informacion del campamento....... \n");

for(i=0;i<7;i++)

{

}

}

else

{

printf(" La Clave introducida no existe \n");

}

Pausa();

}

void ConsultaDatosParticipante()

{

}

void ListadoDeporte()

{

}

void ImportesTotales()

{

}

void PromedioPlazas()

{

}

int Menu\_informes()

{

int opcion;

printf("\n\n \t\t\t\tMENU DE INFORMES");

printf("\n\t\t\t\t----------------- \n\n");

printf("\t\t\t\t1.- Listado General de Campamentos.\n\n");

printf("\t\t\t\t2.- Consulta de datos de un Campamento.\n\n");

printf("\t\t\t\t3.- Consulta de los datos de un Participante.\n\n");

printf("\t\t\t\t4.- Listado por Deporte.\n\n");

printf("\t\t\t\t5.- Importes totales.\n\n");

printf("\t\t\t\t6.- Promedio de Plazas.\n\n");

printf("\t\t\t\t0.- Volver al Menu Principal.\n\n");

scanf("%i",&opcion);

fflush(stdin);

return(opcion);

}

//---------------------------------------------------------------------------------------------------------------------

void MensajeDespedida()

{

printf("\n\n\n\n\t\t\t\t GRACIAS POR UTILIZAR NUESTRO PROGRAMA!\n\n\n\n\n\n\n");

system("pause\n\n");

}

void Pausa() //subprograma pausa

{

system("pause");

}

void AltaCampamentos(struct Campamento \*Campamento){ //peticion y validacion de la clave

char Clave[8];

int i, clavecorrecta, claveigual=1;

do {

printf("introduce la clave de alta de campamentos: \n");

fflush(stdin);

gets(Clave);

clavecorrecta = validar\_clave(Clave);

if(clavecorrecta){

claveigual=comprobar\_clave\_igual(&\*Campamento,&\*Clave);

if(claveigual==0)

strcpy(Campamento->clave,Clave);

}

else{

printf(" \n se ha producido un error , introduzca un formato de clave valido \n ");

Pausa();

}

}while((clavecorrecta==0)||(claveigual==1));

system("cls");

printf(" \n\n INTRODUZCA LA INFORMACION DEL CAMPAMENTO \n");

printf("----------------------------------------------- \n\n");

printf("\t numero maximo de plazas o asistentes que habra en el campamento, (min 5 y max 10) habiles en el campamento: \n");

scanf("%i",&Campamento->Plazas);

printf("\t rango de edad minimo comprendido: \n");

scanf("%i",&Campamento->RangoEdadMin);

printf("\t rango de edad maximo comprendido: \n");

scanf("%i",&Campamento->RangoEdadMax);

printf("\t cual sera el precio del campamento: \n");

scanf("%i",&Campamento->Precio);

printf("\t primera semana de duracion del campamennto: \n");

scanf("%i",&Campamento->Semana1);

printf("\t segunda semana de duracion del campamennto: \n");

scanf("%i",&Campamento->Semana2);

printf("\t que mes se realizara: \n ");

scanf("%s",&Campamento->Mes);

fflush(stdin);

printf("\tjj donde tendra lugar el campamento: \n");

scanf("%s",&Campamento->Lugar);

printf("\n\t\t\t EL CAMPAMENTO HA SIDO DADO DE ALTA CORRECTAMENTE, gracias. \n");

Pausa();

system("cls");

}

int RegistroParticipantes(struct Campamento \*Campamento){

char clave\_participantes[8];

int i;

int opcion=0;

tRegisPlazas auxPlazas;

printf("\n introduce la clave del campamento en el que quieres registrar el participante \n");

gets(clave\_participantes);

for (i=0;(i<7)&& strcmp((Campamento+i)->clave,clave\_participantes);i++);

if (i<7)

{

//printf("\n la clave introducida es valida \n");

do{

printf(" \n NOMBRE DEL PARTICIPANTE: \n");

scanf("%s",&auxPlazas.Nombre);

printf("\n numeros del DNI (solo numeros) del participante: \n");

scanf("%i",& auxPlazas.DNInum );

printf("\n letra del DNI(solo letra) del participante: \n");

scanf("%s",&auxPlazas.DNIletra);

printf(" \n edad del participante(edad cuando asiste al campamento): \n");

scanf("%i",&auxPlazas.Edad);

printf("\n sexo del participante (V o H) \n");

scanf("%s",&auxPlazas.Sexo);

printf("\n desea introducir mas participantes? (S=1/N=0) \n");

scanf("%i",&opcion);

(Campamento+i)->Asistentes[(Campamento+i)->ocupadas]=auxPlazas;

(Campamento+i)->ocupadas++;

}while(((Campamento+i)->ocupadas<(Campamento+i)->Plazas)&&(opcion=1));

// if((Campamento+i)->Plazas=(Campamento+i)->ocupadas)

//else

// {

// printf(" \n ATENCION no quedan plazas libres en este campamento \n");

// }

}

else

{

printf("\n no existe la clave introducida, por favor, inserte una clave asociada a un campamento \n");

}

}

int MenuInformes(){

int opcion;

struct Campamento campamentos[7];

do{

opcion=Menu\_informes();

system("cls");

switch(opcion){

case 1:

ListadoCampamentos(campamentos);

break;

case 2:

ConsultaCampamentos(campamentos);

break;

case 3:

ConsultaDatosParticipante(campamentos);

break;

case 4:

ListadoDeporte();

break;

case 5:

ImportesTotales();

break;

case 6:

PromedioPlazas();

break;

case 0:

system("cls");

break;

default : printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR FAVOR, INGRESE UN NUMERO VALIDO.\n ");

} // fjn del switch

}while (opcion!=0);

}

# *PRUEBAS DE EJECUCIÓN:*

