

Evaluación de Aprendizaje número 03
Apellido y Nombre: Falvo Mathias
Comisión nro.: 63 Turno: Mañana
Materia: Elementos de Programación

Fecha 28/06/2020
1ra. Entrega
DNI: 40345608
Días Cursada: jueves y sábado
Código:1024

Enunciado : Tema "C"

Somos una empresa que dicta cursos de informática. Deseamos tener un sistema para controlar la inscripción de los alumnos a los mismo. El menú principal del sistema deberá tener las distintas opciones:

- 1- : Carga de Cursos
- 2- : Carga de Inscripción
- 3- : Listar Ocupación de Cursos
- 4- : Generar Reportes.
- 5- : Salir

A continuación, se detalla que debe realizar cada una de las opciones:

1- Carga de Cursos El Instituto puede trabajar como máximo, 20 cursos distintos (porque cuenta solamente con 20 aulas).

Por cada uno de los cursos se ingresa:

- ✓ Código de Curso (int , entre 100 y 1000).
- ✓ Descripción de Curso (char, 50 caracteres)
- ✓ Capacidad (int). (hasta 50 alumnos)
- ✓ Precio por curso (float)

Los Cursos se deben guardar en un archivo llamado CursoOfe.dat para que no deban ser cargados cada vez que se inicia el programa. El Programa debe permitir grabar un archivo nuevo, como también continuar con la carga. Finaliza la carga con Curso igual a 99.

2- Carga de Inscripción.

Los Alumnos realizan la solicitud para inscribirse a los cursos a través de la siguiente información:

- ✓ Apellido y Nombre (char, 60 caracteres).
- ✓ Dirección de Email (char, 50 caracteres).
- ✓ Código de Curso (int, entre 100 y 1000)
- ✓ Cantidad (int)

Las solicitudes se deben guardar en un archivo llamado Incripcion.dat, debe confirmar la existencia del código de curso en archivo CursoOfe.dat (función Buscar). Si está el Curso, tiene que descontar de Capacidad. Si la cantidad solicitada es mayor a la disponible o ya no tiene más vacantes, se debe generar un archivo con la misma estructura de Incripcion.dat, bajo el nombre de Sin_Inscrip.dat, para luego enviarle un aviso de las novedades a los Inscriptos. Si no lo encuentra sólo mandar mensaje por pantalla, Curso inexistente. Termina la carga de solicitudes con Código de Cursos igual a 0.

3- Listar Ocupación de Cursos Mostrar por pantalla un listado de los montos de los cursos que tengan inscriptos:

Descripción de Curso	Inscriptos	monto
.NET MVC	20	50000

4- Generar Reportes.

Leer el Archivo Error_Inscripcion.dat: Mostrar por pantalla

Apellido y Nombre	Dirección de Email	Cod Curso	Cantidad
XXXXXXXXXXXXXXXXX	XXXXXX@xxxxxxx	9999	99

5- Salida Permite cerrar el programa.

Consideraciones:

- Cada vez que se selecciona una opción del menú se ejecutan las acciones y luego se vuelve a mostrar el menú hasta que el usuario decida salir y cerrar el programa.
- Tenga en cuenta de que hay acciones que no se pueden realizar si antes no se hicieron otras

Código

```
#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

typedef struct
{
    int cod;

    char descri[50];

    int cap;
```

```
float price;  
}cursos;
```

```
typedef struct
```

```
{  
    char name[60];  
    char email[50];  
    int cod;  
    int cant;  
}solicitudes;
```

```
void leer_Texto(char [], int );  
int menuPrincipal();  
int contarRegistros(char[] );  
int buscarCod(int, char[] );  
void cargaCursos(char );  
int buscarCap(int , char []);  
void descontarCap(int , char [], int );  
void cargaInscripcion();  
int contarInscri(int );  
void cursosInscriptos();  
void noInscriptos();
```

```
int main()
```

```
{  
    int opc, valido;  
    char carga;
```

```
    opc = 0;
```

```
    while (opc != 5)
```

```

{
opc = menuPrincipal();
system("cls");
switch (opc)
{
case 1:
do
{
printf("\n [N] - Crear archivo nuevo");
printf("\n [C] - Continuar con la carga");
printf("\n\n Elija una opcion : ");
fflush(stdin);
scanf("%c", &carga);
fflush(stdin);
carga = toupper(carga);
} while (carga != 'N' && carga != 'C');
cargaCursos(carga);
system("cls");
break;

case 2:
cargaInscripcion();
system("cls");
break;

case 3:
cursosInscriptos();
system("pause");
system("cls");
break;

```

case 4:

```
noInscriptos();  
system("pause");  
system("cls");  
break;
```

case 5:

```
printf("\033[0;31m");  
printf("\n Hasta luego ! \n\n");  
printf("\033[0m");  
break;
```

default:

```
printf("\n Ingrese una opcion correcta.");  
system("pause");  
break;
```

```
system("cls");
```

```
}
```

```
}
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
void leer_Texto(char texto[], int largo)
```

```
{
```

```
int i = 0;
```

```
fflush(stdin);
```

```
fgets(texto,largo,stdin);
```

```

while(texto[i] != '\0')
{
    if(texto[i] == '\n')
    {
        texto[i] = '\0';
    }
    else
    {
        i++;
    }
}
}

```

```

int menuPrincipal()
{
    int opc;

    printf("\033[0;32m");
    printf("\n \t\t\t\t --Menu principal--\n\n");
    printf("\033[0m");
    printf("\n [1] - Carga de cursos");
    printf("\n [2] - Carga de inscripcion");
    printf("\n [3] - Listar ocupacion de cursos");
    printf("\n [4] - Generar reportes");
    printf("\n [5] - Salir");
    do
    {
        printf("\n\n Elija una opcion : ");
        scanf("%d",&opc);
        fflush(stdin);
        if(opc < 1 || opc > 5)

```

```

    {
        printf("\n Asegurese de ingresar una opcion valida.\n");
    }
} while (opc < 1 || opc > 5);

return opc;
}

int contarRegistros(char name[]) // Contador de registros dentro de un archivo.
{
    int cont, read;
    cursos curso;
    FILE *arch;
    arch = fopen("CursoOfe.dat", "rb");
    if(arch == NULL)
    {
        printf("\n Se produjo un error al abrir el archivo.");
        system("pause");
        exit(1);
    }
    cont = 0;

    while (!feof(arch))
    {
        read = fread(&curso, sizeof(cursos), 1, arch);
        cont++;
    }
    fclose(arch);
    return cont;
}

```

```
int buscarCod(int cod, char name[]) // Busca el codigo y muestra si ya existe o no
```

```
{
    int existe, read;
    cursos curso;
    FILE *arch;

    arch = fopen(name,"rb");
    if(arch == NULL)
    {
        printf("\n Se produjo un error al abrir el archivo.");
        system("pause");
        exit(1);
    }
    existe = 0;

    while(!feof(arch) && existe == 0)
    {
        read = fread(&curso,sizeof(cursos),1,arch);
        if(cod == curso.cod)
        {
            existe = 1;
        }
    }
    fclose(arch);
    return existe;
}
```

```
void cargaCursos(char opcion) // Primera opcion del menu
```

```
{
    FILE *carga, *inscrip, *sInscrip;
    int cont, existe;
```



```
char modo[5];
```

```
cursos curso;
```

```
if(opcion == 'N')
```

```
{
```

```
    strcpy(modo,"wb");
```

```
    cont = 0;
```

```
    inscrip = fopen("Inscripcion.dat","wb");
```

```
    fclose(inscrip);
```

```
    sInscrip = fopen("Sin_Inscrip.dat","wb");
```

```
    fclose(sInscrip);
```

```
}
```

```
else
```

```
{
```

```
    strcpy(modo,"ab");
```

```
    cont = contarRegistros("CursoOfe.dat");
```

```
}
```

```
carga = fopen("CursoOfe.dat",modo);
```

```
if(carga == NULL)
```

```
{
```

```
    printf("\n Se produjo un error al abrir el archivo.");
```

```
    system("pause");
```

```
    exit(1);
```

```
}
```

```
do
```

```
{
```

```
    printf("\n Ingrese el codigo del curso entre 100 y 1000 (o 99 para finalizar la carga) :");
```

```
    scanf("%d",&curso.cod);
```

```
    fclose(carga);
```

```

existe = buscarCod(curso.cod, "CursoOfe.dat");
carga = fopen("CursoOfe.dat", "ab");
if(existe)
{
    printf("\n Este codigo de curso ya ha sido utilizado, intentelo nuevamente!");
}
} while (curso.cod < 99 || curso.cod > 1000 || existe == 1);

while(curso.cod != 99 && cont <= 20)
{
    printf("\n Descripcion :");
    leer_Texto(curso.descri, 50);

    do
    {
        printf("\n Capacidad :");
        scanf("%d", &curso.cap);
    } while (curso.cap > 50 || curso.cap < 1);

    printf("\n Precio : $");
    scanf("%f", &curso.price);

    cont++;
    fwrite(&curso, sizeof(cursos), 1, carga);

    //Nuevo comienzo de ciclo
    do
    {
        printf("\n Ingrese el codigo del curso entre 100 y 1000 (o 99 para finalizar la carga) :");
        scanf("%d", &curso.cod);
        fclose(carga);
    }

```

```

    existe = buscarCod(curso.cod, "CursoOfe.dat");

    carga = fopen("CursoOfe.dat", "ab");

    if(existe)
    {
        printf("\n Este codigo de curso ya ha sido utilizado, intentelo nuevamente!");
    }

    } while (curso.cod < 99 || curso.cod > 1000 || existe == 1);
}

fclose(carga);
}

```

int buscarCap(int cod, char name[]) // Busca la capacidad maxima por curso y la retorna

```

{
    int capacidad, read;
    cursos curso;
    FILE *arch;

    arch = fopen(name, "rb");
    if(arch == NULL)
    {
        printf("\n Se produjo un error al abrir el archivo.");
        system("pause");
        exit(1);
    }
}

```

capacidad = 0;

while(!feof(arch) && !capacidad)

```

{
    read = fread(&curso, sizeof(cursos), 1, arch);
    if(cod == curso.cod)

```

```

    {
        capacidad = curso.cap;
    }
}
fclose(arch);
return capacidad;
}

```

void descontarCap(int cod, char name[], int cant) // Descuenta las vacantes libres del curso

```

{
    int read, write, copiado;
    cursos curso;
    FILE *arch, *aux;

    arch = fopen(name,"rb");
    if(arch == NULL)
    {
        printf("\n Se produjo un error al abrir el archivo.");
        system("pause");
        exit(1);
    }
}

```

```

aux = fopen("Auxiliar.dat","wb");
if(arch == NULL)
{
    printf("\n Se produjo un error al abrir el archivo.");
    system("pause");
    exit(1);
}

```

//Lee el archivo y lo copia en un aux modificandolo

```

copiado = 0;
while(!feof(arch))
{
    read = fread(&curso,sizeof(cursos),1,arch);
    if(cod == curso.cod && !copiado)
    {
        copiado = 1;
        curso.cap -= cant;
        write = fwrite(&curso,sizeof(cursos),1,aux);
    }
    else
    {
        if(read)
        {
            write = fwrite(&curso,sizeof(cursos),1,aux);
        }
    }
}
fclose(aux);
fclose(arch);

// Copia el archivo ya modificado al principal
arch = fopen(name,"wb");
if(arch == NULL)
{
    printf("\n Se produjo un error al abrir el archivo.");
    system("pause");
    exit(1);
}
aux = fopen("Auxiliar.dat","rb");
if(arch == NULL)

```

```

{
    printf("\n Se produjo un error al abrir el archivo.");
    system("pause");
    exit(1);
}
while(!feof(aux))
{
    read = fread(&curso,sizeof(cursos),1,aux);
    if(read)
    {
        write = fwrite(&curso,sizeof(cursos),1,arch);
    }
}
fclose(arch);
fclose(aux);
}

```

void cargaInscripcion() //Segunda opcion del menu

```

{
    int cod, existe, cant, cap, write;
    char name[60], email[50];
    solicitudes estudiante;
    FILE *inscri, *sInscri;

    do
    {
        printf("\n Ingrese el codigo del curso (o 0 para finalizar) : ");
        scanf("%d", &cod);
        if(cod != 0)
        {
            existe = buscarCod(cod,"CursoOfe.dat");

```

```

        if(!existe)
        {
            printf("\n Curso inexistente, intentelo nuevamente ! \n");
        }
    }
} while (!existe && cod != 0 && (cod >= 100 || cod <= 1000));

```

```

while(cod != 0)
{
    printf("\n Apellido y nombre : ");
    leer_Texto(name,60);
    printf("\n Email : ");
    leer_Texto(email,50);
    printf("\n Cantidad de vacantes a reservar : ");
    fflush(stdin);
    scanf("%d", &cant);
    cap = buscarCap(cod,"CursoOfe.dat");
    //Carga de campos
    estudiante.cod = cod;
    estudiante.cant = cant;
    strcpy(estudiante.name,name);
    strcpy(estudiante.email,email);
    //Condiciona para escritura de archivo
    if(cap - cant >= 0)
    {
        descontarCap(cod,"CursoOfe.dat",cant);
        inscri = fopen("Inscripcion.dat", "ab");
        write = fwrite(&estudiante,sizeof(solicitudes),1,inscri);
        fclose(inscri);
        printf("\n Felicidades, inscripcion realizada con exito! ");
    }
}

```

```

else
{
    sInscri = fopen("Sin_Inscrip.dat", "ab");
    write = fwrite(&estudiante,sizeof(solicitudes),1,sInscri);
    fclose(sInscri);

    printf("\n Lo sentimos, la cantidad de vacantes disponibles en el curso es = %d. ", cap);
}

//Nuevo inicio
do
{
    printf("\n Ingrese el codigo del curso (o 0 para finalizar) : ");
    scanf("%d", &cod);
    if(cod != 0)
    {
        existe = buscarCod(cod,"CursoOfe.dat");
        if(!existe)
        {
            printf("\n Curso inexistente, intentelo nuevamente ! \n");
        }
    }
} while (!existe && cod != 0 && (cod >= 100 || cod <= 1000));
}
}

```

int contarInscri(int cod) // Cuenta la cantidad de inscriptos por codigo de curso

```

{
    int cantInscri, read;
    solicitudes estudiante;
    FILE *arch;

    arch = fopen("Inscripcion.dat","rb");

```



```

if(arch == NULL)
{
    printf("\n Se produjo un error al abrir el archivo.");
    system("pause");
    exit(1);
}

cantInscri = 0;
while(!feof(arch))
{
    read = fread(&estudiante,sizeof(solicitudes),1,arch);
    if(cod == estudiante.cod && read == 1)
    {
        cantInscri += estudiante.cant;
    }
}
fclose(arch);
return cantInscri;
}

void cursosInscriptos()
{
    int cantInscri, read, inscripto;
    float monto;
    FILE *arch;
    cursos curso;

    printf("\n Descripcion \t\t\t Inscriptos \t\t Monto\n");
    printf("-----");

    arch = fopen("CursoOfe.dat", "rb");

```

```

if(arch == NULL)
{
    printf("\n Se produjo un error al abrir el archivo.");
    system("pause");
    exit(1);
}

inscripto = 0;
while(!feof(arch))
{
    read = fread(&curso,sizeof(cursos),1,arch);
    cantInscri = contarInscri(curso.cod);
    if(cantInscri > 0 && read == 1)
    {
        monto = (float)cantInscri * curso.price;
        printf("\n %10s \t\t %8d \t\t $%.2f", curso.descri, cantInscri, monto);
        inscripto = 1;
    }
}

if(!inscripto)
{
    printf("\n Aun no se cuenta con ningun alumno inscripto.");
}

printf("\n");
fclose(arch);
}

```

```

void noInscriptos()
{
    int flag, read;
    FILE *sInscrip;

```

solicitudes estudiante;

```
sInscrip = fopen("Sin_Inscrip.dat","rb");
printf("\n \t\t--Alumnos que no pudieron registrarse-- \n\n");
printf("\n Apellido y nombre \t\t\t Direccion de Email \t\t\t Cod. Curso \t\t Cantidad");
printf("\n -----
--- \n");

flag = 0;
while(!feof(sInscrip))
{
    read = fread(&estudiante,sizeof(solicitudes),1,sInscrip);
    if(read)
    {
        flag = 1;

        printf("\n %20s \t\t\t %20s \t\t\t %6d \t\t %6d", estudiante.name, estudiante.email,
estudiante.cod, estudiante.cant);

    }
}
if(!flag)
{
    printf("No hay ningun alumno sin inscribir !");
}
printf("\n");
}
```