

Práctica de Archivos

"Archivos de texto"

1.
 - a) La función `close()` ¿utiliza el nombre externo de un archivo, o la variable asociada a éste?
 - b) ¿y la función `fprintf()` ?
 - c) ¿Cuál es la función que usa ambos nombres, el externo y el de la variable asociada?
2. Siendo `arch` una variable de archivo (`FILE`):
 - a) explique la diferencia entre:

```
arch = fopen("archivo.txt", "rt");
```

```
arch = fopen("archivo.txt", "wt");
```
 - b) ¿Presenta algún peligro potencial la instrucción:

```
arch = fopen("archivo.txt", "wt");
```

 ? ¿Porqué?
3.
 - a) Suponga que tiene un archivo *prog1.ent*. Escriba las instrucciones necesarias para abrir este archivo para lectura, y para cerrarlo después.
 - b) Escriba las instrucciones para abrir otro archivo llamado *prog1.sal*, de manera que se pueda escribir en él y luego cerrarlo.
4. Suponiendo que el archivo *alum.dat* contiene nombre y calificación promedio de los alumnos de un colegio, complete las partes en blanco del programa "abanderados", para que imprima los nombres de aquellos alumnos que obtuvieron al menos 9 de calificación promedio.

/* Programa para determinacion de los abanderados */

```
#include <stdio.h>
```

```
main()
```

```
{char nombre[31];
```

```
int nota;
```

```
_____ archa;
```

```
archa = fopen("alum.dat",_____)
```

```
while (_____)
```

```
{    fscanf(_____, "%s", _____);
```

```
    fscanf(_____, "%d", &_____);
```

```
    if (nota >= 9)
```

```
        printf("%s\n", nombre);}
```

```
}
```

5. Escribe un programa para crear un archivo *divis.dat* que contenga todos los divisores de un número entero que se ingresa por teclado. Cada divisor debe estar en una línea diferente.
6. Se posee en un disco un archivo de texto llamado *alumnos.dat*. Cada grupo de datos en el archivo posee el nombre de un alumno y sus tres calificaciones. Escribe un programa para calcular y mostrar en pantalla el nombre de cada alumno y su promedio.
7. Repite el ej.6, pero suponiendo que el archivo *alumnos.dat* posee un encabezado que especifica cuántos alumnos hay en el archivo.
8. Escribe un programa para actualizar, mensualmente, el archivo de sueldos (*sueldos.dat*) de una empresa. Cada grupo de datos en el archivo contiene el nombre del empleado, el sueldo semanal, y finalmente los meses y años de antigüedad en la empresa. Al fin del mes, se debe incrementar la antigüedad en meses (y en años si corresponde), y deben darse aumentos. Si

el empleado tiene al menos 5 años, recibe un aumento del 30%, y si supera los 10 años, se le aumenta un 35% adicional.

9. Cada grupo de datos para el archivo clientes.dat contiene el nombre del cliente, dirección, sexo y saldo de la cuenta en un supermercado. Escribe un programa que:
 - a) Cree dos archivos separados, *mujer.dat* y *varon.dat* con todos los datos de los clientes almacenados en el archivo correcto;
 - b) Muestre en pantalla la siguiente información:
 - Número total de clientes
 - Número de varones
 - Número de mujeres
 - Promedio de saldos para mujeres
 - Promedio de saldos para hombres.

"Archivos binarios"

1. Escribe la declaración del tipo estructurado que se guardará en un archivo de datos de los socios del club "San Antonio de Padua". Cada registro contendrá los siguientes campos (elementos de la estructura):
 - nombre
 - sexo
 - edad
 - fecha último pago de abono.
2. Para el archivo del ejercicio anterior realiza un programa que permita crear el archivo y guardar registros. Construye una función que abra el archivo para lectura, muestre sus registros en pantalla y cierre el archivo.
3. El objeto del siguiente fragmento de programa es actualizar la edad de la socia que esta en el segundo registro del archivo (registro #1), a 29 años. ¿Hay algún error en el proceso? ¿Cómo se solucionaría?

.....

```
typedef struct
```

```
{
```

```
.....
```

```
} CLUB;
```

```
CLUB socio;
```

```
FILE *arch;
```

```
arch = fopen(.....);
```

```
fseek(arch,2,SEEK_SET);
```

```
fread((void*)&socio, sizeof(CLUB),1,arch);
```

```
socio.edad = 34;
```

```
fwrite((void*)&socio, sizeof(CLUB),1,arch);
```

```
.....
```

4. Escribe un programa que:

- a) Almacene en un archivo binario de enteros los números entre 100 y 200.
 - b) Escribe otro programa que almacene en un archivo binario de enteros los números pares entre 200 y 800.
4. Un archivo binario contiene el nombre y el salario mensual para cada uno de los empleados de la compañía ADE. Escriba un programa que incremente el salario de cada empleado en un 10%, actualizando el archivo.
5. El archivo binario estudia.dat , contiene registros (estructuras) con los siguientes datos de los estudiantes de una universidad:

- Nombre
- Número de Legajo
- Cantidad de materias aprobadas
- Promedio de calificaciones

El archivo se encuentra ordenado en orden ascendente por número de legajo. El programa debe presentar un menú con las siguientes opciones:

- a) Mostrar todos los datos de un estudiante, cuyo nro. de legajo se ingresa por teclado.
- b) Mostrar todos los datos de los estudiantes cuyo promedio sea mayor o igual al ingresado por el usuario, y además que tengan no menos de 10 materias aprobadas.

NOTA: Previamente deberá realizar un programa auxiliar para la creación del archivo de estudiantes.

Práctica Integradora

1. Explica las características y forma de uso de los archivos de texto. Presentar un ejemplo explicado.
2. Explica las características y forma de uso de los archivos binarios. Presentar un ejemplo explicado.
3. Explica las siguientes funciones, aportando ejemplos de cada una:
 - a. feof()
 - b. fscanf()
 - c. fseek()
 - d. fread()
4. Escribe un programa que lea valores de un archivo de vectores de 10 números enteros e imprima por pantalla la media de cada vector. (El archivo se ha creado utilizando fprintf).
5. Escribe un programa que copie en un archivo otros dos archivos, uno a continuación de otro.
6. Dado un archivo de texto, escribir un programa que cuente el número de palabras que aparezcan en dicho archivo.
7. Se tienen en un listado los precios de cierre de varias acciones para cada día de la semana pasada. Construye un programa que cargue en un vector o matriz los datos del archivo y que imprima por cada acción los precios máximos y mínimos y el día en que se produjeron.
8. Tenemos un archivo de registros cuyas componentes están definidas según la declaración siguiente:

```
struct registro1{  
    char partido[60];  
    char localidad[60];
```

```
int candidatos;
}
```

Se quiere sustituir la estructura del registro por el siguiente:

```
struct registro1{
char partido[60];
char localidad[60];
int candidatos,elegidos;
}
```

Escribe un programa que modifique la estructura y traslade toda la información del archivo primitivo al nuevo, inicializando a 0 el nuevo campo creado.

9. Dada una palabra, diremos que está “escondida” en un texto si está contenida en éste, no necesariamente de forma continua. Por ejemplo, la palabra “elefante” está escondida en la frase “El hermano de Francisco tuvo suerte”. Elabora un programa tal que, dada una palabra leída por teclado, la busque en un texto almacenado en un archivo de texto texto.txt. Como resultado, el programa debe exponer el mensaje:

Palabra <palabra> no encontrada en el texto

o bien

Palabra <palabra> encontrada escondida entre los <p> primeros caracteres del texto

En el ejemplo, el resultado debería ser:

Palabra elefante encontrada escondida entre los 32 primeros caracteres del texto

Considera la equivalencia entre mayúsculas y minúsculas. Supondremos que la palabra a buscar no tiene más de 20 caracteres, pero desconocemos su longitud.

10. En un archivo de texto glosario.dat se encuentran una serie de abreviaturas en orden alfabético seguidas por su descripción en la siguiente línea. Diseña y elaborar un programa que dándole una abreviatura del archivo conteste con su descripción completa. Si la abreviatura pedida no apareciese en el archivo, el programa indicará la abreviatura anterior y posterior. **Nota:** Las abreviaturas tienen una longitud máxima de 10 caracteres, pero desconocemos la longitud máxima de las descripciones.
11. Dado un valor entero por teclado, elaborar un programa que busque en un archivo (numeros.dat) de enteros ese valor, y en caso de encontrarlo, que diga en qué posición o posiciones del archivo se encuentra. El archivo de enteros ha sido grabado haciendo uso de la función fwrite(), por lo que se hace necesario recuperar los valores elemento a elemento haciendo uso de la función fread().
12. Se define el tipo siguiente:

```
struct alumno {
char apellidos[60];
char nombre[20];
float notajunio,notasept,notaprac;}
}
```

En un archivo con datos de esta estructura disponemos de los alumnos matriculados en una asignatura. Se quiere obtener un archivo de texto en el que aparezca en cada línea el nombre de un alumno con el formato apellidos, nombre, nota junio. Hacer un programa que realice esta tarea.

13. En un archivo se tiene grabada una sucesión de números enteros positivos o nulos, correspondientes a las puntuaciones de varios jueces sobre un mismo ejercicio. Escribe un algoritmo para calcular la media aritmética de los valores estrictamente positivos, excepto el máximo y el mínimo de dichos valores positivos (suponiendo que sólo hay un máximo y un mínimo). También debe calcular el número de ceros. Varía el algoritmo para el caso en el que los valores máximo y mínimo puedan aparecer varias veces.

14. Escribe un programa que permita llevar los gastos e ingresos menores a \$25000 de una empresa. El programa presentará un menú como el siguiente:

```
CONTROL DE GASTOS (versión 1.0)
=====
1. Ingresos
2. Gastos
3. Salida
=====
```

Donde las funciones realizan las siguientes operaciones:

- *menú* muestra el menú en la pantalla.
 - *lee_opcion* lee la opción del teclado.
 - *ingresos* lee nuevos ingresos y los suma al saldo actual.
 - *gastos* lee nuevos gastos y los resta del saldo actual (que nunca podrá ser negativo).
 - *salida* muestra el saldo actual (ingresos - gastos).
15. Dados dos archivos binarios y secuenciales: "A.DAT" y "B.DAT" con el siguiente formato de registros:
- ```
struct Treg {
 int Clave: integer;
 char Dato[10];}
```

Ambos archivos están clasificados en forma ascendente por el campo Clave y dentro de un mismo archivo no hay claves duplicadas.

Escribe cada uno de las siguientes funciones:

- a) Union: se genera un archivo C resultado de la unión entre A y B, es decir el archivo C contendrá todos los registros consignados en los archivos de entrada. En caso de existir claves repetidas en el Archivo A y el B, se grabará un solo registro en C utilizando los datos del archivo B.
  - b) Intersección: se generará el archivo C conteniendo solo los registros del Archivo A cuyas claves también aparezcan en el Archivo B.
  - c) Diferencia: el archivo C tendrá los registros que estén en el primer archivo que se pase como parámetro y cuyas claves no aparezcan en el archivo que se pase como segundo parámetro.
  - d) Genera un archivo "D.DAT" ordenado por clave actualizando los datos que están en A y en B, teniendo en cuenta:
    1. Ambos archivos están ordenados por clave. Las claves en A no se repiten pero las claves en B se pueden repetir
    2. Todos los registros de A serán grabados en D.
    3. Los registros de B se grabarán en D solo si no tienen igual por clave en el archivo A.
    4. Si en B existieran varios registros con una misma clave, que no está en el archivo A, solo se grabará en D el primero de los registros con esa clave leído en B.
16. Un club cuenta con un archivo maestro de acceso secuencial denominado SOCIOS.DAT. Este archivo se encuentra ordenado por código de socio y tiene los siguientes campos:
- Código de socio

- Nombre y apellido
- Dirección
- Deuda (array de 12 posiciones, cada posición contendrá la deuda correspondiente a ese mes)

Se cuenta además con un archivo BAJAS.DAT de acceso **secuencial ordenado**, con los códigos de socio a ser dados de baja. Se desea:

- Actualizar el archivo SOCIOS con el de BAJAS.
- Emitir un listado ordenado por código de socio con: **código de socio, nombre y apellido, dirección y total adeudado** de aquellos socios que hayan sido dados de baja.

Se supone que no hay duplicados en el archivo Socios, pero si puede haberlo en el de Bajas.

Se cuenta además con un archivo Pagos.DAT de acceso **secuencial**, con los datos de los pagos realizados por los socios. Este archivo tiene registros con la siguiente estructura:

- Código de socio
- Mes
- Monto abonado

Este archivo está ordenado en forma ascendente por código de socio. Se desea:

- Actualizar el archivo SOCIOS con la información del archivo Pagos.
- Emitir un listado ordenado por código de socio con: **código de socio, nombre y apellido, total adeudado** con aquellos socios que mantengan deudas con el club.

Se supone que no hay duplicados en el archivo Socios, pero si puede haber varios registros para un mismo socio en el archivo de pagos.

17. Dados dos archivos binarios "A.DAT" (acceso directo y secuencial) y "B.DAT" (acceso secuencial) con el siguiente formato de registros:

```
struct Treg {
 int Clave: integer;
 char Dato[10];
}
```

Se desea realizar un algoritmo que actualice los datos del archivo A con los del B. Teniendo en cuenta que la clave en A coincide con el número de registro. Toda clave en B estará en A.

Se desea realizar un algoritmo que actualice los datos del archivo A con los del B. Teniendo en cuenta que la clave en A **no** coincide con el número de registro. Toda clave en B estará en A.

18. Se cuenta con los siguientes archivos:

MAESTRO (de acceso secuencial ordenado en forma descendente por nro. de cliente)

- Número de cliente (entero)
- Nombre y apellido (string de 40 posiciones)
- Código categoría (entero, de 0 a 99)
- Deuda (array de enteros de 12 posiciones, una por cada mes, conteniendo la deuda mensual en cantidad de KW. El mes que no se adeude tendrá cargado un cero en esa posición.)

CATEGORIAS (de acceso directo por **código de categoría**)

- Código categoría (entero, de 0 a 99)
- Valor KW (real)

NOVEDADES (secuencial ordenado en forma descendente por **nro. de cliente**)

- Número de cliente (entero)
- Mes a pagar (entero)

Se desea:

- a) Actualizar el archivo MAESTRO mediante el archivo NOVEDADES.
- b) Generar un listado ordenado por **nro de cliente** sólo con las NOVEDADES con el siguiente formato:
  - Número de cliente
  - Mes pagado
  - Código categoría
  - Cantidad KW
  - Valor KW
  - Total (cantidad KW \* valor KW)
  - TOTAL RECAUDADO (sólo al finalizar el listado)

#### Aclaraciones

- Los archivos se encuentran cargados y con datos válidos.
- El archivo NOVEDADES puede contener más de una novedad por código de cliente.
- Los archivos de organización secuencial deberán ser tratados exclusivamente de esa forma, lo mismo que los directos serán tratados únicamente en forma directa.

19. Desarrollar un algoritmo que permita la actualización de un archivo de solicitudes de planes de ahorro para compra de automóviles. Se cuenta con:

- ✓ un archivo secuencial ordenado en forma ascendente por el campo NRO-SOLICITUD donde están registradas todas las solicitudes vigentes. Los registros de este archivo tienen la siguiente estructura:

#### REGISTRO-SOLICITUD

|               |                                                    |
|---------------|----------------------------------------------------|
| NRO-SOLICITUD | entero                                             |
| TITULAR       | cadena de 40 caracteres                            |
| MARCA         | cadena de 20 caracteres (Son 12 marcas diferentes) |
| MODELO        | carácter (válidos: A,B,C,D)                        |
| COLOR         | entero (1 a 8)                                     |

- ✓ Y un archivo también secuencial que informa las bajas a procesar. Este archivo está ordenado en forma ascendente por el campo NRO-SOLICITUD-BAJA y sus registros tienen la siguiente estructura:

#### REGISTRO-BAJA

|                    |                  |
|--------------------|------------------|
| NRO-SOLICITUD-BAJA | entero           |
| CAUSA              | carácter (R,F,O) |

- ✓ Se quiere generar un nuevo archivo de solicitudes resultado de efectuar las bajas correspondientes. (Este archivo tendrá la misma estructura que el archivo de solicitudes.)
- ✓ Se podrán rechazar bajas por causa inválida, o baja a registro inexistente. Las bajas rechazadas serán listadas indicando el porque del rechazo.
- ✓ Al final del proceso se emitirá un informe indicando para cada marca, la cantidad de solicitudes existentes por modelo y color.
- ✓ También al final del proceso se indicará la cantidad de registros leídos en cada uno de los archivos de entrada, la cantidad de registros grabados en el archivo actualizado, la cantidad de bajas rechazadas y listadas y la/s marca/s con mayor cantidad de solicitudes vigentes (en el actualizado).