

Trabajo Práctico N° 1: **Creación, Consulta y Mantenimiento de Archivos** **Secuenciales - Algorítmica Básica.**

Ejercicio 1.

Realizar un algoritmo que cree un archivo de números enteros no ordenados y permita incorporar datos al archivo. Los números son ingresados desde teclado. La carga finaliza cuando se ingresa el número 30000, que no debe incorporarse al archivo. El nombre del archivo debe ser proporcionado por el usuario desde teclado.

```
program TP1_E1;
{$codepage UTF8}
uses crt;
const
    num_salida=30000;
type
    t_archivo=file of int16;
procedure leer_numero(var num: int16);
var
    i: int8;
begin
    i:=random(10);
    if (i=0) then
        num:=num_salida
    else
        num:=random(high(int16));
    end;
end;
procedure cargar_archivo(var archivo: t_archivo);
var
    num: int16;
begin
    rewrite(archivo);
    textcolor(green); write('Los números ingresados son: ');
    leer_numero(num);
    while (num<>num_salida) do
    begin
        textcolor(yellow); write(num, ' ');
        write(archivo,num);
        leer_numero(num);
    end;
    close(archivo);
end;
var
    archivo: t_archivo;
    nombre: string;
begin
    randomize;
    nombre:='TP1_E1';
    assign(archivo,nombre);
    cargar_archivo(archivo);
end.
```

Ejercicio 2.

Realizar un algoritmo que, utilizando el archivo de números enteros no ordenados creado en el Ejercicio 1, informe por pantalla cantidad de números menores a 1500 y el promedio de los números ingresados. El nombre del archivo a procesar debe ser proporcionado por el usuario una única vez. Además, el algoritmo deberá listar el contenido del archivo en pantalla.

```
program TP1_E2;
{$codepage UTF8}
uses crt;
const
    num_corte=1500;
type
    t_archivo=file of int16;
procedure procesar_archivo(var archivo: t_archivo; var nums_corte: int16; var prom: real);
var
    num: int16;
    suma: real;
begin
    reset(archivo);
    suma:=0;
    textcolor(green); write('El contenido del archivo es: ');
    while (not eof(archivo)) do
        begin
            read(archivo,num);
            textcolor(yellow); write(num,' ');
            if (num<num_corte) then
                nums_corte:=nums_corte+1;
            suma:=suma+num;
        end;
    if (filesize(archivo)>0) then
        prom:=suma/fileSize(archivo);
    writeln();
    close(archivo);
end;
var
    archivo: t_archivo;
    nums_corte: int16;
    prom: real;
    nombre: string;
begin
    nombre:='TP1_E1';
    nums_corte:=0; prom:=0;
    assign(archivo,nombre);
    procesar_archivo(archivo,nums_corte,prom);
    textcolor(green); write('La cantidad de números menores a '); textcolor(yellow);
write(num_corte); textcolor(green); write(' es '); textcolor(red); writeln(nums_corte);
    textcolor(green); write('El promedio de los números ingresados es '); textcolor(red);
write(prom:0:2);
end.
```

Ejercicio 3.

Realizar un programa que presente un menú con opciones para:

(a) Crear un archivo de registros no ordenados de empleados y completarlo con datos ingresados desde teclado. De cada empleado, se registra: número de empleado, apellido, nombre, edad y DNI. Algunos empleados se ingresan con DNI 00. La carga finaliza cuando se ingresa el String “fin” como apellido.

(b) Abrir el archivo anteriormente generado y

(i) Listar en pantalla los datos de empleados que tengan un nombre o apellido determinado, el cual se proporciona desde el teclado.

(ii) Listar en pantalla los empleados de a uno por línea.

(iii) Listar en pantalla los empleados mayores de 70 años, próximos a jubilarse.

Nota: El nombre del archivo a crear o utilizar debe ser proporcionado por el usuario.

Ejercicio 4.

Agregar al menú del programa del Ejercicio 3 opciones para:

(a) Añadir uno o más empleados al final del archivo con sus datos ingresados por teclado. Tener en cuenta que no se debe agregar al archivo un empleado con un número de empleado ya registrado (control de unicidad).

(b) Modificar la edad de un empleado dado.

(c) Exportar el contenido del archivo a un archivo de texto llamado "todos_empleados.txt".

(d) Exportar a un archivo de texto llamado "faltaDNIEmpleado.txt" los empleados que no tengan cargado el DNI (DNI en 00).

Nota: Las búsquedas deben realizarse por número de empleado.

Ejercicio 5.

Realizar un programa para una tienda de celulares, que presente un menú con opciones para:

(a) Crear un archivo de registros no ordenados de celulares y cargarlo con datos ingresados desde un archivo de texto denominado “celulares.txt”. Los registros correspondientes a los celulares deben contener: código de celular, nombre, descripción, marca, precio, stock mínimo y stock disponible.

(b) Listar en pantalla los datos de aquellos celulares que tengan un stock menor al stock mínimo.

(c) Listar en pantalla los celulares del archivo cuya descripción contenga una cadena de caracteres proporcionada por el usuario.

(d) Exportar el archivo creado en el inciso (a) a un archivo de texto denominado “celulares.txt” con todos los celulares del mismo. El archivo de texto generado podría ser utilizado en un futuro como archivo de carga (ver inciso (a)), por lo que debería respetar el formato dado para este tipo de archivos en la Nota 2.

Nota 1: El nombre del archivo binario de celulares debe ser proporcionado por el usuario.

Nota 2: El archivo de carga debe editarse de manera que cada celular se especifique en tres líneas consecutivas. En la primera, se especifica: código de celular, precio y marca, en la segunda, stock disponible, stock mínimo y descripción; y, en la tercera, nombre en ese orden. Cada celular se carga leyendo tres líneas del archivo “celulares.txt”.

Ejercicio 6.

Agregar al menú del programa del Ejercicio 5 opciones para:

- (a) Añadir uno o más celulares al final del archivo con sus datos ingresados por teclado.*
- (b) Modificar el stock de un celular dado.*
- (c) Exportar el contenido del archivo binario a un archivo de texto denominado "SinStock.txt" con aquellos celulares que tengan stock 0.*

Ejercicio 7.

Realizar un programa que permita:

(a) *Crear un archivo binario a partir de la información almacenada en un archivo de texto. El nombre del archivo de texto es: “novelas.txt”. La información en el archivo de texto consiste en: código de novela, nombre, género y precio de diferentes novelas argentinas. Los datos de cada novela se almacenan en dos líneas en el archivo de texto. La primera línea contendrá la siguiente información: código novela, precio y género; y la segunda línea almacenará el nombre de la novela.*

(b) *Abrir el archivo binario y permitir la actualización del mismo. Se debe poder agregar una novela y modificar una existente. Las búsquedas se realizan por código de novela.*

Nota: El nombre del archivo binario es proporcionado por el usuario desde el teclado.