

Program alumnos;

{
Considere un archivo de tipo TEXT donde cada línea tiene un formato fijo con los siguientes elementos: un número de LU, la cantidad (N) de notas de un alumno, seguido de una secuencia de N enteros que representan sus notas, y finalmente un texto con nombre y apellido (separados por un guión).
Ejemplo de un posible archivo de texto:

```
=====
102020 3 9 7 8  Jose - Prado
102123 4 9 10 6 7 Maria Laura - Ambar
102034 2 7 9  Mara - Selena
102004 1 8 Pedro - Bohn
=====
```

Problema propuesto:

Escriba un procedimiento que dado un número de libreta, si lo encuentra muestre nombre, apellido y el promedio del alumno. Realice un programa de prueba.

Solución:

El problema fue dividido en tres primitivas:

- + procedimiento buscar_alumno
- + función calcular_promedio
- + procedimiento mostrar_nombres

Algoritmo:

Una vez obtenidos los datos de entrada: LU y archivo de texto con los datos el procedimiento "buscar_alumno" hace lo siguiente:

Recorre el archivo de texto desde el comienzo.

Mientras no llegue al final del archivo o a encontrar el LU

hacer: leer un número de libreta

 Si el número leído es el dato de entrada LU

 entonces: calcular promedio (otra primitiva)

 mostrar nombre (otra primitiva)

 Mostrar promedio

 de lo contrario: saltar al final de la línea del archivo

Si no encuentra lo que busca muestra un mensaje en pantalla

}

VAR {.....variables globales.....}

datos: text; {el manejador del archivo de texto con los datos de alumnos}

lu: integer; {el número de libreta a buscar en el archivo}

op:char; {una letra como opción para seguir repitiendo el programa}

{-----}

procedure buscar_alumno(lu: integer; var datos: text);

{

Este procedimiento recibe como entrada un nro de libreta (lu) y un manejador de archivo de texto (datos) con el formato indicado arriba.

El procedimiento tiene dos primitivas auxiliares en su entorno local:

}

{...variables locales de "buscar_alumno"....}

VAR nro: integer; promedio:real; encontrado: boolean;

```

{.....}
function calcular_promedio ( var datos: text):real;
{La función recibe un manejador de un archivo
 que se asume abierto para lectura y listo para leer un número entero
 que representa la cantidad de notas de un alumnos (mayor que 0)}
var i,suma, cant, nro: integer;
begin
  read(datos,cant); {primero lee la cantidad de notas del alumno}
  suma:=0;
  for i:=1 to cant do {luego suma todas las notas del alumno}
  begin
    read(datos, nro); suma:= suma + nro;
  end;
  calcular_promedio:= suma / cant; {retorna el promedio}
end; {fin de la función "calcular_promedio"}
{.....}
procedure mostrar_nombres( var datos: text);
{Recibe un manejador que asume abierto para lectura
 y listo para leer nombres y apellido. Lee caracter a caracter del archivo
 y muestra por pantalla hasta que llega al final de una línea del archivo.}
var ch: char;
begin
  write(' Nombre: '); ch:=' ';
  while not eoln(datos) and (ch <> '-') do
  begin write(ch); read(datos,ch) end;
  writeln; {NO muestra el guión que separa los nombres del apellido}
  write(' Apellido: '); ch:=' ';
  while not eoln(datos) do begin write(ch); read(datos,ch) end;
  writeln;
end; {fin del procedimiento "mostrar_nombres"}
{.....}
begin {comienzo del procedimiento buscar_alumno}
reset(datos); encontrado:=false;
while not eof(datos) and not encontrado do
begin {recorro el archivo hasta encontrar o llegar al final si es que no está}
  read(datos,nro);  writeln(' Elemento leído:', nro);
  if nro=lu then {encontró el alumno que busco}
  begin encontrado:=true; {cambio encontrado a true para salir del while}
    promedio:=calcular_promedio(datos); {llamo a la primitiva}
    mostrar_nombres(datos);           {llamo a la primitiva}
    writeln('El promedio de sus notas es ', promedio:0:2);
  end
  else readln(datos); {si no es el alumno buscado salto al final de la línea}
end; {fin del while}
if not encontrado
  then writeln('El nro de LU ingresado no está presente en el archivo');
close(datos);
end; {fin del procedimiento buscar_alumno}
{-----}
begin {comienzo del programa alumnos}
assign(datos, 'alumnos.txt');
repeat {el programa se repetirá hasta que el usuario ingrese "n"}
  write('Ingrese un nro de LU ');
  readln(lu);
  buscar_alumno(lu,datos);
  writeln;
  write('¿Quiere ejecutar nuevamente la búsqueda? ');
  readln(op);
until op='n'; {si ingresa "n" termina el programa}
end.

```