



# Examen de Teoría Soluciones

19 de junio de 2024

**Fundamentos de Informática**  
Grado en Tecnologías Industriales

*Duración: 3 horas*

## Propuesta de solución:

```
1 program turrify;
2 const
3   MAXTRK = 32;
4   MAXALB = 100000;
5   MAXPLT = 100;
6
7 type
8   tpTrack = record
9     tit : string;
10    dur : real;
11    path : string;
12  end;
13   tpAlbum = record
14     tit : string;
15     tracks : array[1..MAXTRK] of tpTrack;
16     ntracks : integer;
17  end;
18   tpLibrary = record
19     albums : array[1..MAXALB] of tpAlbum;
20     nalbuns : integer;
21  end;
22
23   tpTrackRef = record
24     album : integer;
25     track : integer;
26  end;
27   tpPlaylist = record
28     tracks : array[1..MAXPLT] of tpTrackRef;
29     ntracks : integer;
30  end;
31
32
33
34
35
36
37
38
39
40
41
42
43
```

```
44 function duracionPlayList(const lb: tpLibrary;
45                             const pl: tpPlaylist): real;
46 var
47   total : real;
48   tr,a,t : integer;
49 begin
50   total := 0;
51   for tr:=1 to pl.ntracks do
52     begin
53       a := pl.tracks[tr].album;
54       t := pl.tracks[tr].track;
55       total := total + lb.albums[a].tracks[t].dur;
56     end;
57   duracionPlayList := total;
58 end;
59
60 procedure creaPlayList(const lb: tpLibrary;
61                         maxdur: real;
62                         var pl: tpPlaylist);
63 var
64   a,t : integer;
65   plfull : boolean;
66 begin
67   pl.ntracks := 0;
68   plfull := false;
69   a := 1;
70   while (a<=lb.nalbuns) and not plfull do
71     begin
72       t := 1;
73       while (t<=lb.albums[a].ntracks) and not plfull do
74         begin
75           if (lb.albums[a].tracks[t].dur <= maxdur)
76             then begin
77               pl.ntracks := pl.ntracks+1;
78               pl.tracks[pl.ntracks].album := a;
79               pl.tracks[pl.ntracks].track := t;
80               plfull := (pl.ntracks=MAXPLT);
81             end;
82           t := t+1;
83         end;
84       a := a+1;
85     end;
86 end;
```

**Propuesta de solución:**

```
1  program GastosExtras;
2
3  const
4      maxTrabajadores = 9;
5      pagoHoraExtra = 20;
6      pagoSalidaCampo = 500;
7      pagoFestivo = 80;
8
9  type
10     tpTrabajador = record
11         horasExtra: real;
12         salidasCampo: integer;
13         festivosTrabajados: integer;
14         antigüedad: integer;
15     end;
16     //también se puede hacer con una matriz y sin el tipo trabajador
17     tpTrabajadores = array[1..maxTrabajadores] of tpTrabajador;
18
19
20 //se asume que ya existe este procedimiento.
21 //la antigüedad se puede calcular a partir de la fila de la matriz, pero he decidido guardarla aquí también
22 function PedirDatos(var trabajadores: tpTrabajadores): integer;
23 var
24     i: integer;
25 begin
26     writeln('A cuántos trabajadores hay que pagarles horas extra?');
27     readln(pedirDatos);
28     for i := 1 to pedirDatos do
29         begin
30             writeln('Introduzca las horas extra del trabajador ', i, ': ');
31             readln(trabajadores[i].horasExtra);
32             writeln('Introduzca las salidas de campo del trabajador ', i, ': ');
33             readln(trabajadores[i].salidasCampo);
34             writeln('Introduzca los festivos trabajados del trabajador ', i, ': ');
35             readln(trabajadores[i].festivosTrabajados);
36             trabajadores[i].antigüedad := i*10;
37         end;
38     end;
39
40
41 procedure CalcularGastoTotal(var trabajadores: tpTrabajadores; n: integer; var totalGastos: real);
42 var
43     i: integer;
44     gastoHorasExtra, gastoSalidasCampo, gastoFestivos, plusAntigüedad: real;
45 begin
46     totalGastos := 0;
47     for i := 1 to n do
48         begin
49             gastoHorasExtra := trabajadores[i].horasExtra * pagoHoraExtra;
50             gastoSalidasCampo := trabajadores[i].salidasCampo * pagoSalidaCampo;
51             gastoFestivos := trabajadores[i].festivosTrabajados * pagoFestivo;
52
53             // Calcular el plus de antigüedad
54             plusAntigüedad := (gastoHorasExtra + gastoSalidasCampo + gastoFestivos) * (trabajadores[i].antigüedad / 100);
55             //plusAntigüedad := (gastoHorasExtra + gastoSalidasCampo + gastoFestivos) * ((i*10) / 100);
56
57             // Sumar al total de gastos
58             totalGastos := totalGastos + gastoHorasExtra + gastoSalidasCampo + gastoFestivos + plusAntigüedad;
59         end;
60     end;
61
62 var
63     trabajadores: tpTrabajadores;
64     totalGastos: real;
65     numTrabajadores: integer;
66
67 begin
68     numTrabajadores:=pedirDatos(trabajadores);
69
70     CalcularGastoTotal(trabajadores, numTrabajadores, totalGastos);
71
72     // Mostrar el resultado
73     writeln('El gasto total del supervisor es: ', totalGastos:0:2, ' euros.');
```

```
74 end.
```

**Propuesta de solución:**

```
1 program cotizaciones;
2
3 const
4     MAXDIAS = 31;
5     MAXHORAS = 24;
6
7 type
8     tpFichReales = file of real;
9
10 var
11     fReales: tpFichReales;
12     fTexto: text;
13     nomFichReales: string;
14
15     cot00, cot: real;
16     sumaDia, cotMediaDia: real; { valdria con una de las dos, se dejan por claridad }
17     sumaDif, mediaDif: real;    { valdria con una de las dos, se dejan por claridad }
18
19     i, metodo, nDias: integer;
20
21     mes, anio: real;
22
23 begin
24
25     // write('Introduce nombre fichero cotizaciones por hora de un mes: ');
26     // readln(nomFichReales);
27     // assign(fReales, nomFichReales);
28     assign(fReales, 'cotizacionesMayo.dat');
29     reset(fReales);
30
31     assign(fTexto, 'cotizaciones.txt');
32     rewrite(fTexto);
33
34     write('Introduce metodo de calculo de cotizacion diaria (1 = 00:00, 2 = promedio): ');
35     readln(metodo);
36
37     read(fReales, mes);
38     read(fReales, anio);
39
40     nDias := 0;
41     sumaDif := 0.0;
42     while not eof(fReales) do
43     begin
44         read(fReales, cot00); { lectura del de las 00:00 del dia }
45         nDias := nDias + 1;
46         sumaDia := cot00;
47         for i:=1 to MAXHORAS-1 do
48         begin
49             read(fReales, cot);
50             sumaDia := sumaDia + cot;
51         end;
52         cotMediaDia := sumaDia / MAXHORAS;
53
54         sumaDif := sumaDif + abs(cotMediaDia - cot00);
55
56         { Escritura en fichero de texto }
57         if (metodo = 1) then {00:00}
58         begin
59             writeln(fTexto, cot00:1:2, ' BTC ', nDias, '-', mes:1:0, '-', anio:1:0)
60         else {promedio}
61         begin
62             writeln(fTexto, cotMediaDia:1:2, ' BTC ', nDias, '-', mes:1:0, '-', anio:1:0);
63         end;
64
65         close(fReales);
66         close(fTexto);
67
68         mediaDif := sumaDif / nDias;
69
70         writeln('Fichero cotizaciones.txt escrito con exito.');
```

```
71     writeln('Se ha utilizado el metodo ', metodo, ' de calculo de cotizacion diaria.');
```

```
72     writeln('La diferencia media entre las cotizaciones de ambos metodos es de: ', mediaDif:1:5);
73 end.
```