

## Trabajo Práctico N° 3: Registros. Ejercicios con Corte de Control.

### Ejercicio 1.

*Dado el siguiente programa:*

```
program TP3_E1;
{$codepage UTF8}
uses crt;
type
  str20=string[20];
  alumno=record
    codigo: integer;
    nombre: str20;
    promedio: real;
  end;
procedure leer(var alu: alumno);
begin
  textcolor(green); write('Introducir código del alumno: '); textcolor(yellow);
  readln(alu.codigo);
  if (alu.codigo<>0) then
  begin
    textcolor(green); write('Introducir nombre del alumno: '); textcolor(yellow);
    readln(alu.nombre);
    textcolor(green); write('Introducir promedio del alumno: '); textcolor(yellow);
    readln(alu.promedio);
  end;
end;
var
  a: alumno;
begin
end.
```

(a) Completar el programa principal para que lea información de alumnos (código, nombre, promedio) e informe la cantidad de alumnos leídos. La lectura finaliza cuando ingresa un alumno con código 0, que no debe procesarse. Nota: utilizar el módulo leer.

```
program TP3_E1a;
{$codepage UTF8}
uses crt;
const
  alumno_salida=0;
type
  str20=string[20];
  alumno=record
    codigo: integer;
    nombre: str20;
    promedio: real;
  end;
procedure leer(var alu: alumno);
begin
  textcolor(green); write('Introducir código del alumno: '); textcolor(yellow);
  readln(alu.codigo);
  if (alu.codigo<>alumno_salida) then
  begin
    textcolor(green); write('Introducir nombre del alumno: '); textcolor(yellow);
    readln(alu.nombre);
```

```

    textcolor(green); write('Introducir promedio del alumno: '); textcolor(yellow);
readln(alu.promedio);
    end;
end;
var
    a: alumno;
    alumnos_leidos: integer;
begin
    alumnos_leidos:=0;
    leer(a);
    while (a.codigo<>alumno_salida) do
        begin
            alumnos_leidos:=alumnos_leidos+1;
            leer(a);
        end;
    textcolor(green); write('La cantidad de alumnos leídos es '); textcolor(red);
write(alumnos_leidos);
end.

```

**(b)** *Modificar al programa anterior para que, al finalizar la lectura de todos los alumnos, se informe también el nombre del alumno con mejor promedio.*

```

program TP3_E1b;
{$codepage UTF8}
uses crt;
const
    alumno_salida=0;
type
    str20=string[20];
    alumno=record
        codigo: integer;
        nombre: str20;
        promedio: real;
    end;
procedure leer(var alu: alumno);
begin
    textcolor(green); write('Introducir código del alumno: '); textcolor(yellow);
readln(alu.codigo);
    if (alu.codigo<>alumno_salida) then
        begin
            textcolor(green); write('Introducir nombre del alumno: '); textcolor(yellow);
readln(alu.nombre);
            textcolor(green); write('Introducir promedio del alumno: '); textcolor(yellow);
readln(alu.promedio);
        end;
    end;
procedure promedios(alu: alumno; var promedio_max: real; var alumno_max: str20);
begin
    if (alu.promedio>promedio_max) then
        begin
            promedio_max:=alu.promedio;
            alumno_max:=alu.nombre;
        end;
    end;
var
    a: alumno;
    alumnos_leidos: integer;
    promedio_max: real;
    alumno_max: str20;
begin
    alumnos_leidos:=0;
    leer(a);

```

```
while (a.codigo<>alumno_salida) do
begin
    alumnos_leidos:=alumnos_leidos+1;
    promedios(a,promedio_max,alumno_max);
    leer(a);
end;
textcolor(green); write('La cantidad de alumnos leidos es '); textcolor(red);
writeln(alumnos_leidos);
textcolor(green); write('El nombre del alumno con mejor promedio es '); textcolor(red);
write(alumno_max);
end.
```

**Ejercicio 2.**

*El registro civil de La Plata ha solicitado un programa para analizar la distribución de casamientos durante el año 2019. Para ello, cuenta con información de las fechas de todos los casamientos realizados durante ese año.*

**(a)** *Analizar y definir un tipo de dato adecuado para almacenar la información de la fecha de cada casamiento.*

```
type
  t_dia=1..31;
  t_mes=1..12;
  t_registro_casamiento=record
    dia: t_dia;
    mes: t_mes;
    anio: int16;
  end;
```

**(b)** *Implementar un módulo que lea una fecha desde teclado y la retorne en un parámetro cuyo tipo es el definido en el inciso (a).*

```
procedure leer_casamientos(var casamientos_verano, casamientos_1a10: int16);
var
  registro_casamiento: t_registro_casamiento;
begin
  leer_casamiento(registro_casamiento);
  while (registro_casamiento.anio<>anio_salida) do
    begin
      if ((registro_casamiento.mes=1) or (registro_casamiento.mes=2) or
(registro_casamiento.mes=3)) then
        casamientos_verano:=casamientos_verano+1;
      if (registro_casamiento.dia<=dia_corte) then
        casamientos_1a10:=casamientos_1a10+1;
      leer_casamiento(registro_casamiento);
    end;
  end;
```

**(c)** *Implementar un programa que lea la fecha de todos los casamientos realizados en 2019. La lectura finaliza al ingresar el año 2020, que no debe procesarse, e informe la cantidad de casamientos realizados durante los meses de verano (enero, febrero y marzo) y la cantidad de casamientos realizados en los primeros 10 días de cada mes. Nota: utilizar el módulo realizado en (b) para la lectura de fecha.*

```
program TP3_E2;
{$codepage UTF8}
uses crt;
const
  anio_salida=2020; dia_corte=10;
type
  t_dia=1..31;
  t_mes=1..12;
  t_registro_casamiento=record
    dia: t_dia;
```

```
mes: t_mes;
anio: int16;
end;
procedure leer_casamiento(var registro_casamiento: t_registro_casamiento);
begin
    textcolor(green); write('Introducir año del casamiento: '); textcolor(yellow);
    readln(registro_casamiento.anio);
    if (registro_casamiento.anio<>anio_salida) then
    begin
        textcolor(green); write('Introducir mes del casamiento: '); textcolor(yellow);
        readln(registro_casamiento.mes);
        textcolor(green); write('Introducir día del casamiento: '); textcolor(yellow);
        readln(registro_casamiento.dia);
    end;
end;
procedure leer_casamientos(var casamientos_verano, casamientos_1a10: int16);
var
    registro_casamiento: t_registro_casamiento;
begin
    leer_casamiento(registro_casamiento);
    while (registro_casamiento.anio<>anio_salida) do
    begin
        if ((registro_casamiento.mes=1) or (registro_casamiento.mes=2) or
(registro_casamiento.mes=3)) then
            casamientos_verano:=casamientos_verano+1;
            if (registro_casamiento.dia<=dia_corte) then
                casamientos_1a10:=casamientos_1a10+1;
            leer_casamiento(registro_casamiento);
        end;
    end;
end;
var
    casamientos_verano, casamientos_1a10: int16;
begin
    casamientos_verano:=0; casamientos_1a10:=0;
    leer_casamientos(casamientos_verano,casamientos_1a10);
    textcolor(green); write('La cantidad de casamientos realizados durante los meses de verano
(enero, febrero y marzo) es '); textcolor(red); writeln(casamientos_verano);
    textcolor(green); write('La cantidad de casamientos realizados en los primeros 10 días de
cada mes es '); textcolor(red); write(casamientos_1a10);
end.
```

### Ejercicio 3.

*El Ministerio de Educación desea realizar un relevamiento de las 2400 escuelas primarias de la provincia de Bs. As., con el objetivo de evaluar si se cumple la proporción de alumnos por docente calculada por la UNESCO para el año 2015 (1 docente cada 23,435 alumnos). Para ello, se cuenta con información de: CUE (código único de establecimiento), nombre del establecimiento, cantidad de docentes, cantidad de alumnos, localidad. Se pide implementar un programa que procese la información y determine:*

- *Cantidad de escuelas de La Plata con una relación de alumnos por docente superior a la sugerida por UNESCO.*
- *CUE y nombre de las dos escuelas con mejor relación entre docentes y alumnos.*

*El programa debe utilizar:*

- *Un módulo para la lectura de la información de la escuela.*
- *Un módulo para determinar la relación docente-alumno (esa relación se obtiene del cociente entre la cantidad de alumnos y la cantidad de docentes).*

```
program TP3_E3;
{$codepage UTF8}
uses crt;
const
  escuelas_total=2400; ratio_unesco=23.435;
type
  registro_escuela=record
    cue: int16;
    nombre: string;
    docentes: int16;
    alumnos: int16;
    localidad: string;
  end;
procedure leer_escuela(var escuela: registro_escuela);
begin
  textcolor(green); write('Introducir CUE (Código Único de Establecimiento) de la escuela: '); textcolor(yellow); readln(escuela.cue);
  textcolor(green); write('Introducir nombre de la escuela: '); textcolor(yellow); readln(escuela.nombre);
  textcolor(green); write('Introducir cantidad de docentes de la escuela: '); textcolor(yellow); readln(escuela.docentes);
  textcolor(green); write('Introducir cantidad de alumnos de la escuela: '); textcolor(yellow); readln(escuela.alumnos);
  textcolor(green); write('Introducir localidad de la escuela: '); textcolor(yellow); readln(escuela.localidad);
end;
function ratio_alumnos_docente(escuela: registro_escuela): real;
begin
  ratio_alumnos_docente:=escuela.alumnos/escuela.docentes;
end;
procedure actualizar_minimos(ratio: real; escuela: registro_escuela; var ratio_min1, ratio_min2: real; var cue_min1, cue_min2: int16; var nombre_min1, nombre_min2: string);
begin
  if (ratio<ratio_min1) then
  begin
    ratio_min2:=ratio_min1;
    cue_min2:=cue_min1;
    nombre_min2:=nombre_min1;
    nombre_min1:=escuela.nombre;
    cue_min1:=escuela.cue;
  end
  else

```

```
    if (ratio<ratio_min2) then
    begin
        ratio_min2:=ratio;
        cue_min2:=escuela.cue;
        nombre_min2:=escuela.nombre;
    end;
end;
procedure leer_escuelas(var escuelas_lp, cue_min1, cue_min2: int16; var nombre_min1,
nombre_min2: string);
var
    escuela: registro_escuela;
    i: int16;
    ratio, ratio_min1, ratio_min2: real;
begin
    ratio:=0; ratio_min1:=999999; ratio_min2:=999999;
    for i:= 1 to escuelas_total do
    begin
        leer_escuela(escuela);
        ratio:=ratio_alumnos_docente(escuela);
        actualizar_minimos(ratio,escuela,ratio_min1,ratio_min2,cue_min1,cue_min2,nombre_min1,nom
bre_min2);
        if ((escuela.localidad='La Plata') and (ratio>ratio_unesco)) then
            escuelas_lp:=escuelas_lp+1;
        end;
    end;
end;
var
    escuelas_lp, cue_min1, cue_min2: int16;
    nombre_min1, nombre_min2: string;
begin
    escuelas_lp:=0; cue_min1:=0; cue_min2:=0;
    leer_escuelas(escuelas_lp,cue_min1,cue_min2,nombre_min1,nombre_min2);
    textcolor(green); write('La cantidad de escuelas de La Plata con una relación de alumnos
por docente superior a la sugerida por UNESCO es '); textcolor(red); writeln(escuelas_lp);
    textcolor(green); write('Los CUEs de las dos escuelas con mejor relación entre docentes y
alumnos son '); textcolor(red); write(cue_min1); textcolor(green); write(' y ');
textcolor(red); writeln(cue_min2);
    textcolor(green); write('Los nombres de las dos escuelas con mejor relación entre docentes
y alumnos son '); textcolor(red); write(nombre_min1); textcolor(green); write(' y ');
textcolor(red); writeln(nombre_min2);
end.
```

**Ejercicio 4.**

*Una compañía de telefonía celular debe realizar la facturación mensual de sus 9300 clientes con planes de consumo ilimitados (clientes que pagan por lo que consumen). Para cada cliente, se conoce su código de cliente y cantidad de líneas a su nombre. De cada línea, se tiene el número de teléfono, la cantidad de minutos consumidos y la cantidad de MB consumidos en el mes. Se pide implementar un programa que lea los datos de los clientes de la compañía e informe el monto total a facturar para cada uno. Para ello, se requiere:*

- *Realizar un módulo que lea la información de una línea de teléfono.*
- *Realizar un módulo que reciba los datos de un cliente, lea la información de todas sus líneas (utilizando el módulo desarrollado en el inciso (a) y retorne la cantidad total de minutos y la cantidad total de MB a facturar del cliente.*

*Nota: para realizar los cálculos tener en cuenta que cada minuto cuesta \$3,40 y cada MB consumido cuesta \$1,35.*



## Ejercicio 5.

Realizar un programa que lea información de autos que están a la venta en una concesionaria. De cada auto, se lee: marca, modelo y precio. La lectura finaliza cuando se ingresa la marca “ZZZ” que no debe procesarse. La información se ingresa ordenada por marca. Se pide calcular e informar:

- El precio promedio por marca.
- Marca y modelo del auto más caro.

```

program TP3_E5;
{$codepage UTF8}
uses crt;
const
    marca_salida='ZZZ';
type
    t_registro_auto=record
        marca: string;
        modelo: string;
        precio: int32;
    end;
procedure leer_auto(var registro_auto: t_registro_auto);
begin
    textcolor(green); write('Introducir marca del auto: '); textcolor(yellow);
    readln(registro_auto.marca);
    if (registro_auto.marca<>marca_salida) then
    begin
        textcolor(green); write('Introducir modelo del auto: '); textcolor(yellow);
        readln(registro_auto.modelo);
        textcolor(green); write('Introducir precio del auto: '); textcolor(yellow);
        readln(registro_auto.precio);
    end;
end;
procedure actualizar_maximos(registro_auto: t_registro_auto; var precio_max: int32; var
marca_max, modelo_max: string);
begin
    if (registro_auto.precio>precio_max) then
    begin
        precio_max:=registro_auto.precio;
        marca_max:=registro_auto.marca;
        modelo_max:=registro_auto.modelo;
    end;
end;
procedure leer_autos(var marca_max, modelo_max: string);
var
    registro_auto: t_registro_auto;
    precio_max, precio_sum, autos_total: int32;
    precio_prom: real;
    marca: string;
begin
    precio_max:=0;
    leer_auto(registro_auto);
    while (registro_auto.marca<>marca_salida) do
    begin
        marca:=registro_auto.marca; precio_sum:=0; autos_total:=0; precio_prom:=0;
        while ((registro_auto.marca=marca) and (registro_auto.marca<>marca_salida)) do
        begin
            precio_sum:=precio_sum+registro_auto.precio;
            autos_total:=autos_total+1;
            actualizar_maximos(registro_auto,precio_max,marca_max,modelo_max);
            leer_auto(registro_auto);
        end;
        precio_prom:=precio_sum/autos_total;
    end;
end;

```

```
    textcolor(green); write('El promedio de la marca '); textcolor(red); write(marca);
textcolor(green); write(' es '); textcolor(red); writeln(precio_prom:0:2);
    end;
end;
var
    marca_max, modelo_max: string;
begin
    leer_autos(marca_max, modelo_max);
    textcolor(green); write('La marca y el modelo del auto más caro son '); textcolor(red);
write(marca_max); textcolor(green); write(' y '); textcolor(red); write(modelo_max);
textcolor(green); write(', respectivamente');
end.
```

## Ejercicio 6.

Una empresa importadora de microprocesadores desea implementar un sistema de software para analizar la información de los productos que mantiene actualmente en stock. Para ello, se conoce la siguiente información de los microprocesadores: marca (Intel, AMD, NVidia, etc.), línea (Xeon, Core i7, Opteron, Atom, Centrino, etc.), cantidad de cores o núcleos de procesamiento (1, 2, 4, 8), velocidad del reloj (medida en Ghz) y tamaño en nanómetros (nm) de los transistores (14, 22, 32, 45, etc.). La información de los microprocesadores se lee de forma consecutiva por marca de procesador y la lectura finaliza al ingresar un procesador con 0 cores (que no debe procesarse). Se pide implementar un programa que lea información de los microprocesadores de la empresa importadora e informe:

- Marca y línea de todos los procesadores de más de 2 cores con transistores de, a lo sumo, 22 nm.
- Las dos marcas con mayor cantidad de procesadores con transistores de 14 nm.
- Cantidad de procesadores multicore (de más de un core) de Intel o AMD, cuyos relojes alcancen velocidades de, al menos, 2 Ghz.

```
program TP3_E6;
{$codepage UTF8}
uses crt;
const
    cores_salida=0; cores_obj=2; transistores_obj1=22; transistores_obj2=14;
    velocidad_obj=2.0;
type
    t_registro_procesador=record
        cores: int16;
        marca: string;
        linea: string;
        velocidad: real;
        transistores: int16;
    end;
procedure leer_procesador(var registro_procesador: t_registro_procesador);
begin
    textcolor(green); write('Introducir cantidad de cores o núcleos de procesamiento del
procesador (1, 2, 4, 8): '); textcolor(yellow); readln(registro_procesador.cores);
    if (registro_procesador.cores<>cores_salida) then
    begin
        textcolor(green); write('Introducir marca del procesador (Intel, AMD, NVidia): ');
        textcolor(yellow); readln(registro_procesador.marca);
        textcolor(green); write('Introducir línea del procesador (Xeon, Core i7, Opteron, Atom,
Centrino): '); textcolor(yellow); readln(registro_procesador.linea);
        textcolor(green); write('Introducir velocidad del reloj (medida en Ghz): ');
        textcolor(yellow); readln(registro_procesador.velocidad);
        textcolor(green); write('Introducir tamaño en nanómetros (nm) de los transistores (14,
22, 32, 45): '); textcolor(yellow); readln(registro_procesador.transistores);
    end;
end;
procedure actualizar_maximos(transistores_marca: int16; marca: string; var
transistores_max1, transistores_max2: int16; var marca_max1, marca_max2: string);
begin
    if (transistores_marca>transistores_max1) then
    begin
        transistores_max2:=transistores_max1;
        marca_max2:=marca_max1;
        transistores_max1:=transistores_marca;
        marca_max1:=marca;
    end
    else
```

```

    if (transistores_marca>transistores_max2) then
    begin
        transistores_max2:=transistores_marca;
        marca_max2:=marca;
    end;
end;
procedure leer_procesadores(var procesadores_intel_amd: int16; var marca_max1, marca_max2:
string);
var
    registro_procesador: t_registro_procesador;
    transistores_marca, transistores_max1, transistores_max2: int16;
    marca: string;
begin
    transistores_max1:=0; transistores_max2:=0;
    leer_procesador(registro_procesador);
    while (registro_procesador.cores<>cores_salida) do
    begin
        marca:=registro_procesador.marca; transistores_marca:=0;
        while ((registro_procesador.marca=marca) and (registro_procesador.cores<>cores_salida))
do
            begin
                if ((registro_procesador.cores>cores_obj) and
(registro_procesador.transistores<=transistores_obj1)) then
                    begin
                        textcolor(green); write('La marca y la línea de este registro_procesador con más de
2 cores con transistores de, a lo sumo, 22 nm. son '); textcolor(red);
write(registro_procesador.marca); textcolor(green); write(' y '); textcolor(red);
writeln(registro_procesador.linea);
                    end;
                    if (registro_procesador.transistores=transistores_obj2) then
                        transistores_marca:=transistores_marca+1;
                    if ((registro_procesador.cores>=cores_obj) and ((registro_procesador.marca='Intel') or
(registro_procesador.marca='AMD')) and (registro_procesador.velocidad>=velocidad_obj)) then
                        procesadores_intel_amd:=procesadores_intel_amd+1;
                        leer_procesador(registro_procesador);
                    end;
                actualizar_maximos(transistores_marca,marca,transistores_max1,transistores_max2,marca_ma
x1,marca_max2);
            end;
        end;
    end;
var
    procesadores_intel_amd: int16;
    marca_max1, marca_max2: string;
begin
    procesadores_intel_amd:=0;
    leer_procesadores(procesadores_intel_amd,marca_max1,marca_max2);
    textcolor(green); write('Las dos marcas con mayor cantidad de registro_procesadores con
transistores de 14 nm. son '); textcolor(red); write(marca_max1); textcolor(green); write('
y '); textcolor(red); writeln(marca_max2);
    textcolor(green); write('La cantidad de registro_procesadores multicore (de más de un
core) de Intel o AMD, cuyos relojes alcancen velocidades de, al menos, 2 Ghz es ');
    textcolor(red); write(procesadores_intel_amd);
end.

```

**Ejercicio 7.**

*Realizar un programa que lea información de centros de investigación de Universidades Nacionales. De cada centro, se lee su nombre abreviado (ej., LIDI, LIFIA, LINTI), la universidad a la que pertenece, la cantidad de investigadores y la cantidad de becarios que poseen. La información se lee de forma consecutiva por universidad y la lectura finaliza al leer un centro con 0 investigadores, que no debe procesarse. Informar:*

- *Cantidad total de centros para cada universidad.*
- *Universidad con mayor cantidad de investigadores en sus centros.*
- *Los dos centros con menor cantidad de becarios.*

**Ejercicio 8.**

*La Comisión Provincial por la Memoria desea analizar la información de los proyectos presentados en el programa Jóvenes y Memoria durante la convocatoria 2020. Cada proyecto, posee un código único, un título, el docente coordinador (DNI, nombre y apellido, email), la cantidad de alumnos que participan del proyecto, el nombre de la escuela y la localidad a la que pertenecen. Cada escuela, puede presentar más de un proyecto. La información se ingresa ordenada consecutivamente por localidad y, para cada localidad, por escuela. Realizar un programa que lea la información de los proyectos hasta que se ingrese el proyecto con código -1 (que no debe procesarse), e informe:*

- *Cantidad total de escuelas que participan en la convocatoria 2018 y cantidad de escuelas por cada localidad.*
- *Nombres de las dos escuelas con mayor cantidad de alumnos participantes.*
- *Título de los proyectos de la localidad de Daireaux cuyo código posee igual cantidad de dígitos pares e impares.*

### **Ejercicio 9.**

*Realizar un programa que lea información de los candidatos ganadores de las últimas elecciones a intendente de la provincia de Buenos Aires. Para cada candidato, se lee: localidad, apellido del candidato, cantidad de votos obtenidos y cantidad de votantes de la localidad. La lectura finaliza al leer la localidad 'Zárate', que debe procesarse. Informar:*

- *El intendente que obtuvo la mayor cantidad de votos en la elección.*
- *El intendente que obtuvo el mayor porcentaje de votos de la elección.*

**Ejercicio 10.**

*Un centro de investigación de la UNLP está organizando la información de las 320 especies de plantas con las que trabajan. Para cada especie, se ingresa su nombre científico, tiempo promedio de vida (en meses), tipo de planta (por ej., árbol, conífera, arbusto, helecho, musgo, etc.), clima (templado, continental, subtropical, desértico, etc.) y países en el mundo donde se las encuentra. La información de las plantas se ingresa ordenada por tipo de planta y, para cada planta, la lectura de países donde se las encuentra finaliza al ingresar el país 'zzz'. Al finalizar la lectura, informar:*

- *El tipo de planta con menor cantidad de plantas.*
- *El tiempo promedio de vida de las plantas de cada tipo.*
- *El nombre científico de las dos plantas más longevas.*
- *Los nombres de las plantas nativas de Argentina que se encuentran en regiones con clima subtropical.*
- *El nombre de la planta que se encuentra en más países.*



**Ejercicio 11.**

*Una compañía de vuelos internacionales está analizando la información de todos los vuelos realizados por sus aviones durante todo el año 2019. De cada vuelo, se conoce el código de avión, país de salida, país de llegada, cantidad de kilómetros recorridos y porcentaje de ocupación del avión. La información se ingresa ordenada por código de avión y, para cada avión, por país de salida. La lectura finaliza al ingresar el código 44. Informar:*

- *Los dos aviones que más kilómetros recorrieron y los dos aviones que menos kilómetros recorrieron.*
- *El avión que salió desde más países diferentes.*
- *La cantidad de vuelos de más de 5.000 km que no alcanzaron el 60% de ocupación del avión.*
- *La cantidad de vuelos de menos de 10.000 km que llegaron a Australia o a Nueva Zelanda.*

## **Ejercicio 12.**

*En la “Práctica 1 - Ejercicios Adicionales”, se resolvieron 3 problemas complejos sin utilizar módulos. Al carecer de herramientas para modularizar, esos programas resultaban difíciles de leer, de extender y de depurar. En la “Práctica 2 (parte 2) - Ejercicios Adicionales”, se adaptaron los 3 problemas para utilizar módulos y, así, organizar mejor el programa. Ahora, podemos incluir los registros y, así, seguir mejorando nuestros programas. Para cada caso, analizar:*

- *¿Qué entidades del programa conviene representar como registros?*
- *¿Qué atributos de cada entidad deben incluirse en los registros?*
- *¿Qué cambios deben realizarse en los módulos implementados en la práctica 2 para aprovechar los nuevos tipos de datos? ¿Conviene seguir utilizando los mismos módulos en todos los casos?*

*Una vez realizado el análisis, modificar los 3 problemas, utilizando registros para representar los datos del programa. Al finalizar cada problema, comparar la solución usando registros y módulos con la solución sin registros y con módulos (Práctica 2) y con la solución sin registros ni módulos (Práctica 1).*

- *¿Qué diferencias observa?*
- *¿Qué similitudes encuentra?*