

PROGRAMACIÓN I

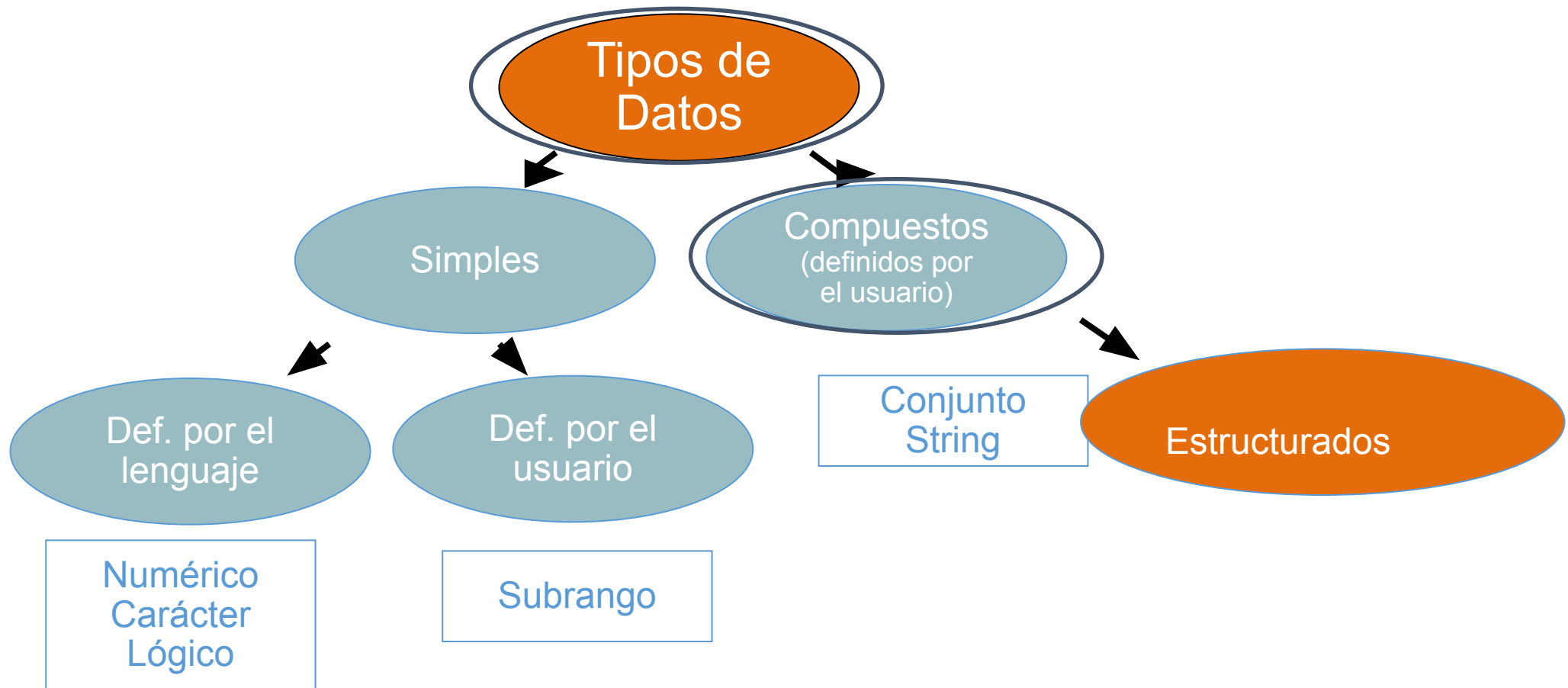
AÑO 2025

1

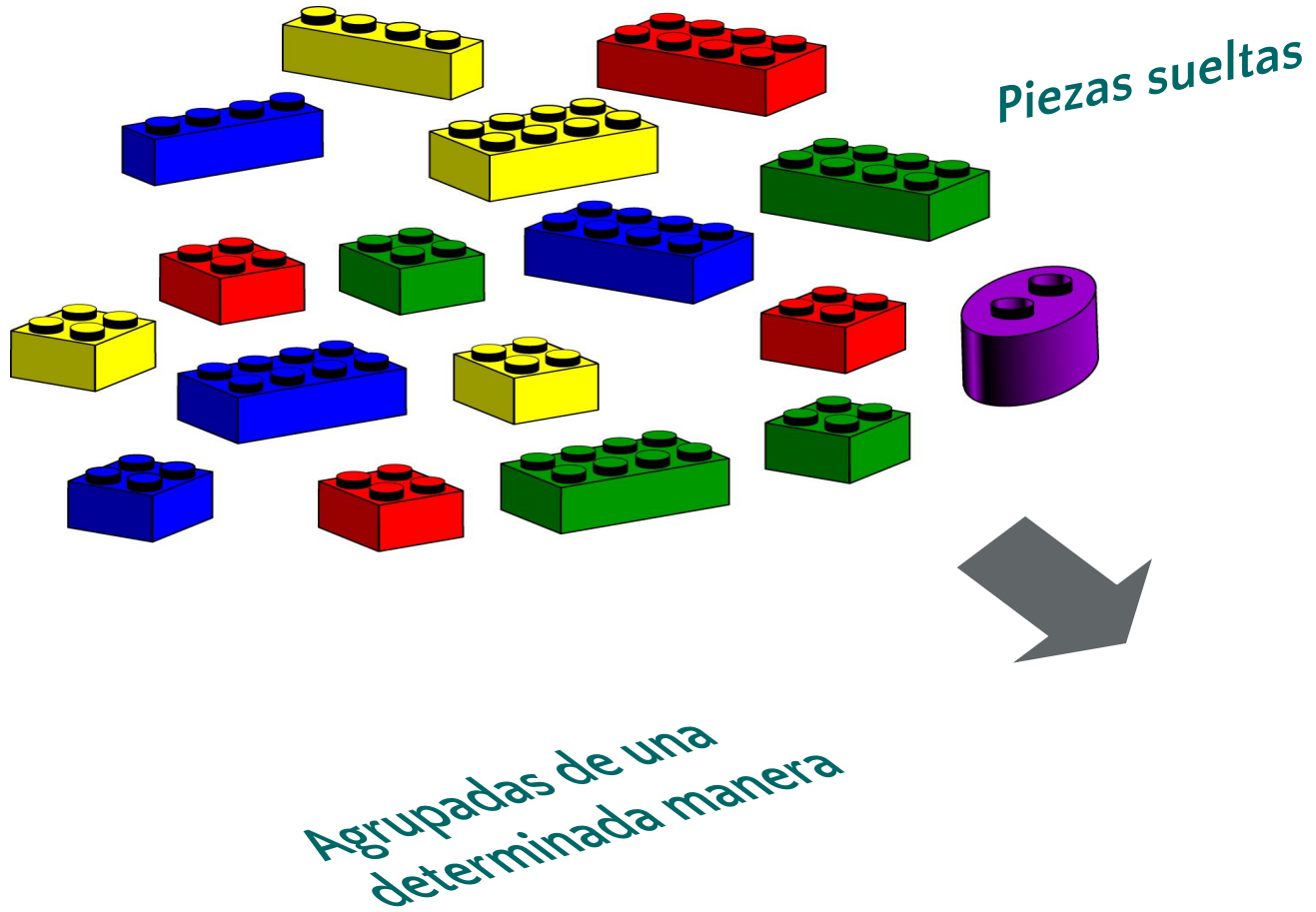
Tipo de dato estructurado/ Estructura de datos

Concepto

Clasificación



Estructuras de datos – Motivación



Mafalda

Estructuras de datos

Una estructura de datos es un conjunto de variables (que podrían ser de distinto tipo) relacionadas entre sí y que se puede operar como un todo, bajo un nombre único.

Esto nos va permitir representar los elementos del mundo real, que generalmente son más complejos que un número entero o una palabra.

Estructuras de datos

Representar los datos de **empleados** de una empresa. Se identifica a través del nombre, el número de documento, la fecha de nacimiento, el número de legajo, el sexo, el sueldo, la antigüedad, etc.



Representar los datos de un **producto del supermercado**. Se debería identificar el código, la marca, la identificación, el precio, la fecha de vencimiento, etc.

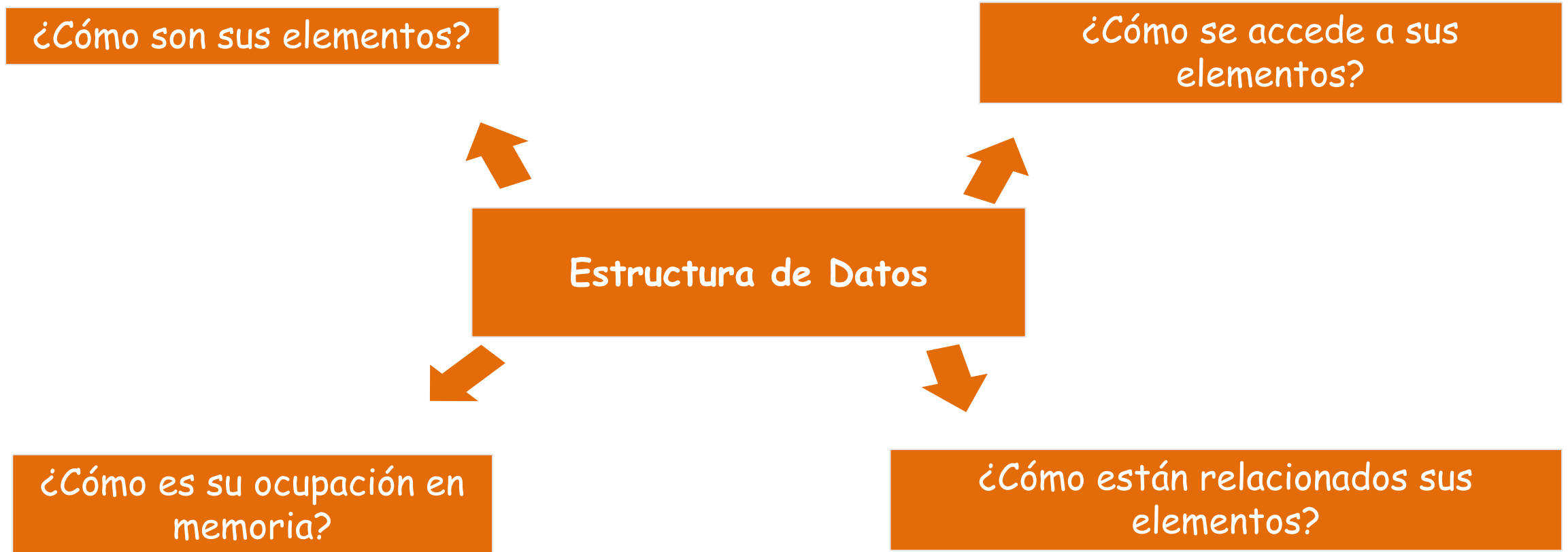


Representar la **lista de clubes de un torneo de fútbol**. Se deberían considerar los nombres de los equipos de fútbol.

Huracán
Independiente
Lanús
San Lorenzo
Arsenal
Banfield
Colón
Newells
Gimnasia (LP)
Estudiantes
Vélez
Olimpo
Quilmes
San Martín (S-J)

Estructuras de datos - Clasificación

Las estructuras de datos se pueden clasificar desde distintos puntos de vista:



Estructuras de datos - Clasificación

De acuerdo a los tipos de datos que se pueden almacenar en la estructura:



Cola de personas

Estructura de Datos

Homogénea

Heterogénea



Los objetos del escritorio



Los Envases Plásticos



Los Animales domésticos



Las partes de mi computadoras

Estructuras de datos - Clasificación

Las estructuras de datos pueden clasificarse, de acuerdo al tipo de datos que la componen, en homogéneas y heterogéneas.

Una estructura de datos se dice **homogénea** si los datos que la componen son todos del mismo tipo.

Una estructura de datos se dice **heterogénea** si los datos que la componen son de distinto tipo.



Cola de personas



Los Envases Plásticos



Los Animales domésticos



Las partes de mi computadoras



Los objetos del escritorio

Estructuras de datos - Clasificación

De acuerdo a la ocupación de memoria las estructuras pueden ser:



Las butacas del teatro

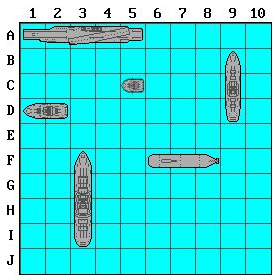
Estructura de Datos

Estática

Dinámica



Una pila de canastos



El cuadro para el juego de la Batalla Naval



La fila de personas frente a la ventanilla de pagos

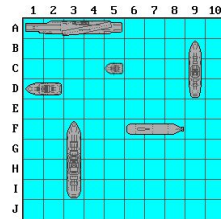
Estructuras de datos - Clasificación

Las estructuras de datos pueden clasificarse, de acuerdo a la ocupación de memoria, en estáticas y dinámicas.

Una estructura de datos se dice **estática** si la cantidad de elementos que contiene es fija, es decir que la cantidad de memoria que ocupa no varía durante la ejecución del programa.



Las butacas del teatro



El cuadro para el juego de la Batalla Naval

Una estructura de datos se dice **dinámica** si la cantidad de elementos que contiene es variable, y por lo tanto la cantidad de memoria ocupada puede cambiar durante la ejecución de un programa.



La pila de canastos



La fila de personas frente a la ventanilla de pagos

Estructuras de datos - Clasificación

De acuerdo al acceso a sus elementos, las estructuras pueden ser:

Microsoft Excel - Libro1

	A	B	C	D
1	ANDRADE GONZALEZ ANA ALICIA			
2	ALMENDRA MENDEZ OSCAR DAVID			
3	CASTILLO VEGA JULIO CESAR			
4	BANDA GARCIA BRENDA			
5	CORZO MORENO LUIS ALBERTO			
6	DIAZ NUÑEZ ISRAEL			
7	CRUZ SANTOS CLAUDIA YOMELI			
8	FLORES ESCOBAR NAHUM			
9	GARCIA COLORADO MARTHA IZTAYANA			
10	HERNANDEZ REYES LUCERO MONTSERRAT			
11	HERNANDEZ ESPEJO KAREN			
12	HERNANDEZ DAVILA RAFAEL			
13	LAGUNES MENDOZA DIANA LAURA			
14	LARA RODRIGUEZ TANIA GUADALUPE			
15	LAGUNES CRUZ XOCHITL JEREMY			
16	LARA SALAS CARLOS ALBERTO			
17	MORALES SILVA RUTH EUNICE			
18	SAGUAYA REYES ANA BARBARA			
19	SOLIS PEREZ ZURISADAI			
20	SALDIVAR ARIAS SANDRA IVONNE			
21	SOTO PORTILLO DENISSE			
22	USCANGA CRUZ IVAN			
23	VELAZQUEZ ZARATE CLAUDIA GIOVANNA			
24	CABRERA MARTINEZ JOSE LUIS			
25	COLIN ROMAN ADILENE VIANEY			
26	REYES CAMPOS ELIZABETH			
27	RODRIGUEZ FUENTES ANA BELEM			
28	SILVA ORTEGA ZELTZIN NAYELI			
29	VASQUEZ SANCHEZ MARIANA			
30	WILGGIN JIMENEZ OMAR			
31				



Estructura de Datos

De acceso secuencial

De acceso directo

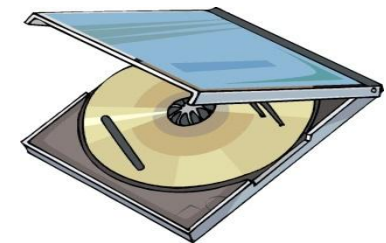


Seleccionar una aplicación determinada

Buscar un nombre en un listado desordenado



Localizar una imagen en un rollo de película



Seleccionar una canción determinada

Estructuras de datos - Clasificación

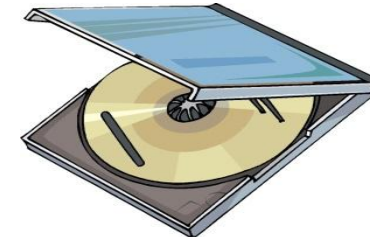
Las estructuras de datos pueden clasificarse, de acuerdo a como se accede a sus elementos, de acceso secuencial o directo.

Una estructura de datos se dice de **acceso secuencial**, si para acceder a un elemento particular se debe respetar un orden predeterminado, por ejemplo, pasando por todos los elementos que le preceden.



Localizar una imagen en un rollo de película

Una estructura de datos se dice de **acceso directo**, si se puede acceder a un elemento particular, directamente, sin necesidad de pasar por los anteriores a él, por ejemplo, indicando una posición.



Seleccionar una canción determinada

Estructuras de datos - Clasificación

De acuerdo a la relación entre sus elementos, las estructuras pueden ser:

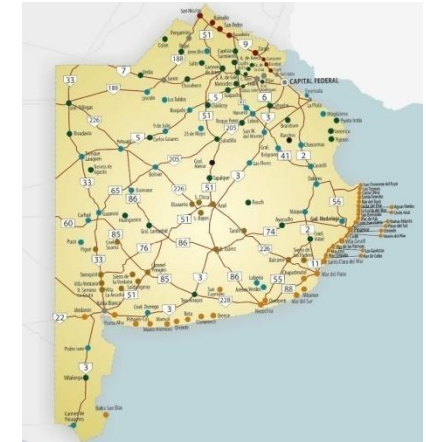


Productos a cobrar en la caja del supermercado

Estructura de Datos

Lineales

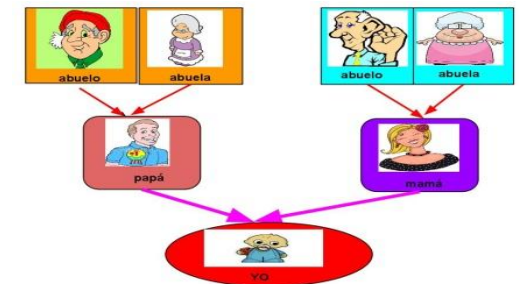
No lineales



Las rutas que unen localidades de la Pcia de Bs. As.



Los vehículos que pasan por la cabina de peaje (cada uno tiene uno antes y uno después)



Nuestro árbol genealógico

Estructuras de datos - Clasificación

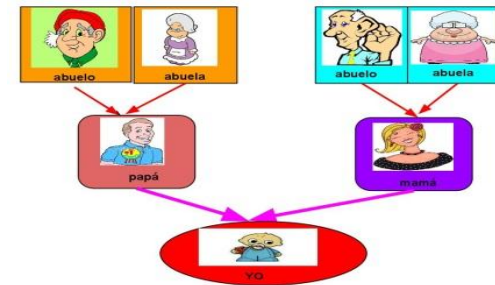
Las estructuras de datos pueden clasificarse, de acuerdo a su relación con sus los otros elementos, en Lineales y No Lineales.

Una estructura de datos se dice **lineal** cuando está formada por ninguno, uno o varios elementos que guardan una relación de adyacencia ordenada donde a cada elemento le sigue uno y le precede uno, **solamente**.



Productos a cobrar en la caja del supermercado

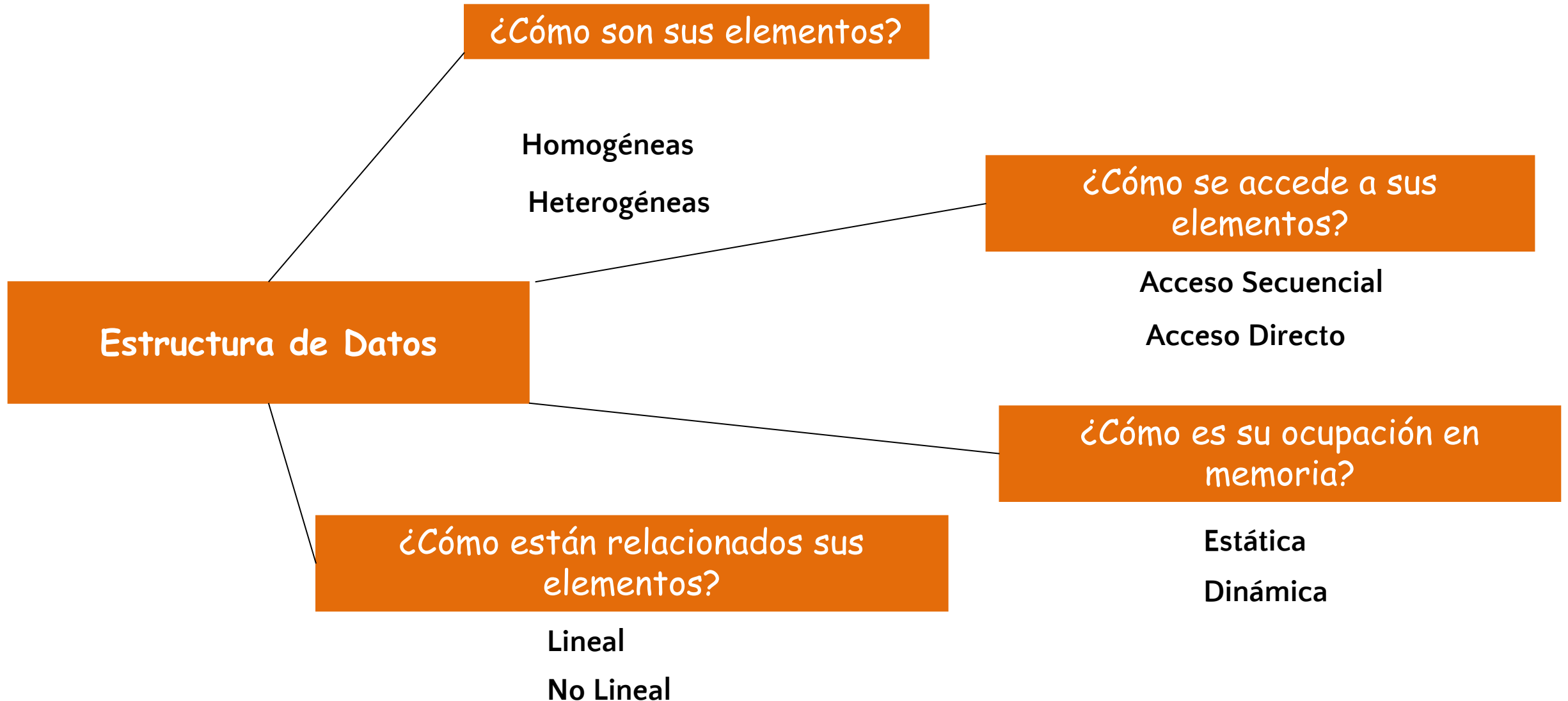
Una estructura de datos se **dice No lineal** si para un elemento dado pueden existir 0, 1 ó mas elementos que le suceden y 0, 1 ó mas elementos que le preceden.



Nuestro árbol genealógico

Estructuras de datos - Resumen

Las estructuras de datos se pueden clasificar desde distintos puntos de vista:



2

Estructura de datos Registro

Tipo de dato Registro – Características

Un registro es una estructura de datos que cumple con:

- Los valores pueden ser de diferentes tipos, esto convierte a un registro en una estructura de datos **heterogénea**
- El almacenamiento ocupado por un registro es fijo, por esto, un registro es una estructura **estática**.
- El **acceso** a sus componentes (campos) es **directo**. Debe referenciarse a través de su nombre.



Nombre
DNI
Fecha Nac.
NroLegajo Sexo
Sueldo
Antigüedad



Nombre
Nro. Alumno
Datos Personales Materias
que cursa
Materias aprobadas



Origen
Destino
Fecha envío Mensaje



Código
Marca
Nombre
Precio
Fecha de vencimiento



Supongamos que se quiere representar la información de los distintos inmuebles que tiene para alquilar una inmobiliaria.



¿Qué información debería conocer la inmobiliaria?



```
Program uno;
```

```
Const
```

```
...
```

```
Type
```

```
inmueble = record
```

```
    tipo: string;
```

```
    cantHab: integer;
```

```
    cantBaños: integer;
```

```
    precio: real;
```

```
    localidad: string;
```

```
end;
```

```
Var
```

```
    inmu1, inmu2: inmueble;
```

*¿Cómo
trabajamos el
registro
inmueble?*



CON LA VARIABLE REGISTRO

```
Program uno;  
Const  
    ....  
Type  
  
    inmueble = record  
        tipo: string;  
        cantHab: integer;  
        cantBaños: integer;  
        precio: real;  
        localidad: string;  
    end;  
  
Var  
    inmu1, inmu2: inmueble;
```

REGISTRO



```
Begin
```

```
    ....  
    inmu2:= inmu1;  
    ...  
End.
```



La única operación permitida es la asignación entre dos variables del mismo tipo



CON LOS CAMPOS DEL REGISTRO

¿Cómo le
damos valor?

```
Program uno;  
Const  
    ....  
Type  
  
    inmueble = record  
        tipo: string;  
        cantHab: integer;  
        cantBaños: integer;  
        precio: real;  
        localidad: string;  
    end;  
Var  
    inmu1, inmu2: inmueble;
```

REGISTRO



Begin

....

Puedo realizar las
operaciones permitidas
según el tipo de campo
del registro

...

End.



La única forma de acceder
a los campos es
variable.nombrecampo
inmu1.cantHab

REGISTRO



```
Program uno;  
Const  
    ....  
Type  
  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;  
  
Var  
    inmu1, inmu2: inmueble;
```

```
Begin  
    inmu1.tipo:='Casa';  
    inmu1.cantHab:= 2;  
    inmu1.cantBaños:= 1;  
    inmu1.precio:= 30000.50;  
    inmu1.localidad:= 'La Plata';  
End.
```

```
Begin  
    read (inmu1.tipo);  
    read(inmu1.cantHab);  
    read(inmu1.cantBaños);  
    read(inmu1.precio);  
    read(inmu1.localidad);  
End.
```

¿Qué ocurre si no le doy valor a todos los campos?

¿Debo asignarlos en el orden en que se declararon?



No se puede hacer read (inmu1)

¿MODULARIZAR?



```
Procedure leerInmueble (var i:inmueble);  
  
Begin  
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);  
end;
```

¿Cómo muestro
el contenido de
un registro?

```
Program uno;  
Const  
    ....  
Type  
    inmueble = record  
        tipo: string;  
        cantHab: integer;  
        cantBaños:integer;  
        precio: real;  
        localidad: string;  
    end;  
Procedure leerInmueble (var i:inmueble);  
begin  
    ....  
end;  
  
Var inmu1, inmu2: inmueble;  
  
Begin  
    leerInmueble (inmu1);  
    inmu2:= inmu1;  
End.
```




```
Program uno;  
Const  
    ....  
Type  
  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;  
  
Var  
    inmu1, inmu2: inmueble;
```

```
Begin  
    leerInmueble (inmu1);  
    write (inmu1.tipo);  
    write(inmu1.cantHab);  
    write(inmu1.cantBaños);  
    write(inmu1.precio);  
    write(inmu1.localidad);  
End.
```



**No se puede hacer
write (inmu1)**

¿MODULARIZAR?



```

Procedure imprimir (i:inmueble);

Begin
    write(i.tipo);
    write(i.cantHab);
    write(i.cantBaños);
    write(i.precio);
    write(i.localidad);
end;

```

```

Program uno;
Const ...
Type inmueble = record
    tipo: string;
    cantHab: integer;
    cantBaños:integer;
    precio: real;
    localidad: string;
end;
Procedure leerInmueble (var i:inmueble);
begin
    ....
end;
Procedure imprimir (i:inmueble);
begin
    ....
end;
var inmu1, inmu2: inmueble;
Begin
    leer (inmu1);
    imprimir(inmu1);
End.

```



```
Program uno;  
Const  
    ....  
Type  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;  
Procedure leerInmueble (var  
    i: inmueble);  
begin  
    ....  
end;  
var inmu1, inmu2: inmueble;
```

Begin

```
    leerInmueble (inmu1);  
    leerInmueble (inmu2),
```



**No se puede hacer
inmu1 = inmu2**

```
    if ((inmu1.tipo = inmu2.tipo) and  
        (inmu1.cantHab = inmu2.cantHab) and  
        (inmu1.cantBaños = inmu2.cantBaños) and  
        (inmu1.precio = inmu2.precio) and  
        (inmu1.localidad = inmu2.localidad)) then  
        write ('Los registros tienen los mismos valores')  
    End.
```

¿MODULARIZAR?



```
function iguales (i1,i2:inmueble):boolean;  
Var  
    ok:boolean;  
Begin  
    if ((i1.tipo = i2.tipo)and  
        (i1.cantHab = i2.cantHab) and  
        (i1.cantBaños = i2.cantBaños) and  
        (i1.precio = i2.precio) and  
        (i1.localidad = i2.localidad))  
    then ok:= true  
    else ok:= false;  
    iguales:= ok;  
end;
```

Otra manera de escribir...

```
function iguales (i1,i2:inmueble):boolean;  
Begin  
    iguales:= ((i1.tipo = i2.tipo)and  
                (i1.cantHab = i2.cantHab) and  
                (i1.cantBaños = i2.cantBaños) and  
                (i1.precio = i2.precio) and  
                (i1.localidad = i2.localidad));  
end;
```



```
Program uno;
Const
    ....
Type
inmueble = record
    tipo: string;
    cantHab: integer;
    cantBaños:integer;
    precio: real;
    localidad: string;
end;
function iguales (i1,i2:inmueble): boolean;
begin
    ....
end;
Procedure leerInmueble (var i:inmueble);
begin
    ....
end;
Var
    inmu1, inmu2: inmueble;
```

```
Begin
    leerInmueble (inmu1);
    leerInmueble (inmu2);
    if (iguales (inmu1,inmu2) = true)
    then write ('Los registros son iguales')
    else write ('Los registros no son iguales');
End.
```

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX` Al finalizar informe de los inmuebles en la localidad de `La Plata` cuantos tienen al menos 2 habitaciones

Tipo 'Depto'
cantHab:2
cantBaños 1
precio 15.200
Localidad 'La Plata'

Tipo 'Casa'
cantHab:3
cantBaños 2
precio 23.000
Localidad 'Gonnet'

Tipo 'Casa'
cantHab:5
cantBaños 3
precio 55.400
Localidad 'La Plata'

Tipo 'Casa'
cantHab:1
cantBaños 2
precio 18.000
Localidad 'La Plata'

Tipo 'Casa'
cantHab:4
cantBaños 1
precio 10.000
Localidad XXX



2

**Al leer la Localidad XXX,
¿necesito leer todos los
otros campos?**

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX` Al finalizar informe de los inmuebles en la localidad de `La Plata` cuantos tienen al menos 2 habitaciones

```
Inicializar contadores (cant)
Leer registro (inmu)
While (no sea el ultimo registro) do
  begin
    if (inmu es de La Plata con al menos dos habitaciones) then
      incremento (cant)
    leer registro (inmu)
  end;
Write ('La cantidad es', cant);
```

*¿Cómo verifico
las condiciones?*

*¿Qué
modularizo?*

EJERCITACIÓN



```
Program uno;  
Const  
    ....  
Type  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;  
  
// módulos  
  
Var  
    inmu1, inmu2: inmueble;  
    cant: integer;
```

```
Begin  
    cant:=0;  
  
    leerInmueble (inmu);  
  
    while (inmu.localidad <> 'XXX') do  
        begin  
            if (cumple (inmu1) = true) then  
                cant:= cant + 1;  
                leerInmueble (inmu);  
            end;  
            write ('La cantidad es', cant);  
        End.
```


EJERCITACIÓN



```
procedure leerInmueble (var i:inmueble);  
Begin  
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);  
end;
```

**¿Qué
alternativa
conviene?**

```
procedure leerInmueble (var i:inmueble);  
Begin  
    read(i.localidad);  
    if (i.localidad <> 'XXX') then  
        begin  
            read(i.cantHab);  
            read(i.cantBaños);  
            read(i.precio);  
            read(i.tipo);  
        end;  
    end;  
end;
```



```
function cumple(i:inmueble): boolean;  
var  
    ok:boolean;  
  
begin  
    if (i.localidad = 'La Plata') and (i.cantHab >= 2)  
    then ok:= true  
    else ok:= false;  
    cumple:= ok;  
end;
```



```
Program uno;
```

```
Const
```

```
....
```

```
Type
```

```
inmueble = record
```

```
    tipo: string;
```

```
    cantHab: integer;
```

```
    cantBaños: integer;
```

```
    precio: real;
```

```
    localidad: string;
```

```
end;
```

```
Var
```

```
    inmu1: inmueble;
```

*¿Qué cambiamos
si ahora se
quiere agregar la
fecha desde que
el inmueble está
disponible?*

Opción 1

```
Program uno;  
Type  
  años = 1980..2050;  
  meses = 1..12;  
  días = 1..31;  
  inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
    día: días;  
    mes: meses;  
    año: años;  
  end;  
Var  
  inmu1: inmueble;
```

REGISTRO



*¿Otra
opción?*

Opción 2

```
Program uno;  
Type  
  años = 1980..2050;  
  meses = 1..12;  
  días = 1..31;  
  fecha = record  
    día: días;  
    mes: meses;  
    año: años;  
  end;  
  inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
    fechaPub: fecha;  
  end;  
end;
```

REGISTRO



*¿Cómo hacemos
ahora el proceso
de lectura?*



```
procedure leerInmueble (var i:inmueble);  
Begin  
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);  
    read (i.fechaPub);  
end;
```

NO SE PUEDE ya que **i.fechaPub** es un registro y no se puede hacer read directamente

```
procedure leerInmueble (var i:inmueble);  
Begin  
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);  
    read (i.fechaPub.dia);  
    read (i.fechaPub.mes);  
    read (i.fechaPub.año);  
end;
```

*¿Otra
alternativa?*



```
procedure leerFecha (var f:fecha);  
Begin  
    read(f.dia);  
    read(f.mes);  
    read(f.año);  
end;
```

```
procedure leerInmueble (var i:inmueble);  
begin  
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);  
    leerFecha (i.fechaPub);  
end;
```

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX` Al finalizar informe la cantidad de inmuebles de la localidad de `La Plata` con al menos dos habitaciones; y la cantidad de inmuebles que se publicaron en el verano del 2020-21.

Localidad 'La Plata'
Tipo 'Depto'
cantHab:2
cantBaños 1
precio 15.200
fechaPub: 13/12/2020

Localidad 'La Plata'
Tipo 'Casa'
cantHab:3
cantBaños 2
precio 23.000
fechaPub: 23/01/2021

Localidad 'Gonnet'
Tipo 'Casa'
cantHab:5
cantBaños 3
precio 55.400
fechaPub: 14/02/2021

Localidad 'La Plata'
Tipo 'Casa'
cantHab:1
cantBaños 2
precio 18.000
fechaPub: 1/04/2021

Localidad XXX



2

2

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX`. Al finalizar informe la cantidad de inmuebles de la localidad de `La Plata` con al menos dos habitaciones; y la cantidad de inmuebles que se publicaron en el verano del 2020-21.

```
Inicializar contadores (cantLP,cantVe)
Leer registro (inmu)
While (no sea el ultimo registro) do
  begin
    if (inmu es de La Plata con al menos dos
habitaciones) then
      incremento (cant)
      if (inmu se publicó en verano 2020-21) then
        incremento (cantVe)
      leer registro (inmu)
    end;
  Write ('Las cantidades son', cant,cantV);
```

*¿Qué
modularizo?*

*¿Cómo verifico
las condiciones?*

Program uno;

Type

anios = 1980..2050;

meses = 1..12;

dias = 1..31;

fecha = record

 dia:dias; mes:meses; año:anios;

end;

inmueble = record

 tipo: string; cantHab: integer;

 cantBaños:integer;

 precio: real; localidad: string;

 fechaPub:fecha;

end;

Procedure leerInmueble (var i:inmueble);

begin

end;

function cumple(i:inmueble): boolean;

begin

end;

function verano (i:fecha): boolean;

begin

end;

EJERCITACIÓN



*¿Cómo es la
función
verano?*

Var

 inmu:inmueble;

 cant,cantVe:integer;

Begin

 cantLP:= 0; cantVe:=0;

 leerInmueble (inmu);

 while (inmu.localidad <> 'XXX') do

 begin

 if (cumple (inmu)= true) then

 cantLP:= cantLP + 1;

 if (**verano (inmu.fechaPub) = true**) then

 cantVe:= cantVe + 1;

 leerInmueble (inmu);

 end;

 write ('Las cantidades son', cantLP,cantVe);

End.

EJERCITACIÓN



```
function verano (f:fecha): boolean;
```

```
Var
```

```
    esVerano:boolean;
```

```
Begin
```

```
    if ( (f.dia>= 21) and (f.mes = 12)
        and (f.año = 2020) ) then esVerano:= true
```

```
    else if ( (f.dia < 21) and (f.mes = 3)
        and (f.año = 2021) ) then esVerano:= true
```

```
    else if ( ( (f.mes = 1) or (f.mes = 2) )
        and (f.año = 2021) ) then esVerano:= true
```

```
    else esVerano:= false;
```

```
verano:= esVerano;
```

```
end;
```

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX` Al finalizar informe el precio de alquiler del inmueble que tiene la máxima cantidad de habitaciones y baños (suma de habitaciones + baños).

Localidad 'La Plata'
Tipo 'Depto'
cantHab:2
cantBaños 1
precio 15.200

Localidad 'Gonnet'
Tipo 'Casa'
cantHab:3
cantBaños 2
precio 23.000

Localidad 'La Plata'
Tipo 'Casa'
cantHab:5
cantBaños 3
precio 55.400

Localidad 'La Plata'
Tipo 'Casa'
cantHab:1
cantBaños 2
precio 18.000

Localidad XXX



55.400

EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX` Al finalizar informe de el precio de alquiler del inmueble que tiene la máxima cantidad de habitaciones y baños (suma de habitaciones + baños).

```
Inicializar máximo (max)
```

```
Leer registro (inmu)
```

```
While (no sea el ultimo registro) do
```

```
begin
```

```
    sumadeHab:= inmu.cantHab + inmu.cantBaños
```

```
    if (sumadeHab supera el máximo) then
```

```
        actualizo el máximo (max)
```

```
        guardo el precio máximo (precioMax)
```

```
    leer registro (inmu)
```

```
end;
```

```
Write ('El precio del inmueble con máxima  
cantidad de habitaciones + baños es', precioMax);
```

**¿Cómo verifico
las condiciones?**

**¿Qué
modularizo?**

EJERCITACIÓN



¿Qué cambiaría del programa si ahora se quiere informar todas las características del inmueble con mayor cantidad de habitaciones + baños?

```
Program uno;
Type
  inmueble = record
    tipo: string;
    cantHab: integer;
    cantBaños: integer;
    precio: real;
    localidad: string;
  end;

Procedure leerInmueble
  (var i: inmueble);
begin
end;

Var
  inmu: inmueble;
  sumadeHab: integer;
  max: integer;
  precioMax: real;
```

```
Begin
  max:= -1;
  leerInmueble (inmu);
  while (inmu.localidad <> 'XXX') do
    begin
      sumadeHab:=inmu.cantHabitaciones + inmu.cantBaños;
      if (sumadeHab >= max) then
        begin
          max:= sumadeHab;
          precioMax:= inmu.precio;
        end;
      leerInmueble (inmu);
    end;
  write ('El precio del inmueble con máxima cantidad
  de habitaciones + baños es', precioMax);
End.
```



```
Program uno;
Type
  inmueble = record
    tipo: string;
    cantHab: integer;
    cantBaños: integer;
    precio: real;
    localidad: string;
  end;

Procedure leerInmueble
  (var i: inmueble);
begin
end;

Var
  inmu: inmueble;
  sumadeHab: integer;
  max: integer;
  tipoMax, locMax: string;
  cantHMax, cantBMax: integer;
  precioMax: real;
```

```
Begin
  max:= -1;
  leerInmueble (inmu);
  while (inmu.localidad <> 'XXX') do
    begin
      sumadeHab:=inmu.cantHabitaciones + inmu.cantBaños;
      if (sumadeHab >= max) then
        begin
          max:= sumadeHab;
          tipoMax:= inmu.tipo;
          cantHMax:= inmu.cantHab;
          cantBMax:= inmu.cantBaños;
          precioMax:= inmu.precio;
          locMax:= inmu.localidad;
        end;
      leerInmueble (inmu);
    end;
  write ('Inmueble con máxima cantidad de
habitaciones + baños es', tipoMax, cantHMax,
cantBMax, precioMax, locMax);
End.
```

**REGISTRO
INMUEBLE**

EJERCITACIÓN



```
Program uno;
Type
inmueble = record
    tipo: string;
    cantHab: integer;
    cantBaños:integer;
    precio: real;
    localidad: string;
end;

Procedure leerInmueble
    (var i:inmueble);
begin
end;

Var
    inmu:inmueble;
    sumadeHab:integer;
    max: integer;
    inmuMax:inmueble;
```

```
Begin
    max:= -1;
    leerInmueble (inmu);
    while (inmu.localidad <> 'XXX') do
        begin
            sumadeHab:=inmu.cantHabitaciones + inmu.cantBaños;
            if (sumadeHab >= max) then
                begin
                    max:= sumadeHab;
                    inmuMax:= inmu;
                end;
            leerInmueble (inmu);
        end;
        write ('Los datos del inmueble con máxima cantidad
de habitaciones + baños son', inmuMax.tipo,
inmuMax.cantHab, inmuMax.cantBaños, inmuMax.precio,
inmuMax.localidad);
    End.
```