



Conceptos de Algoritmos Datos y Programas



CADP – Temas de la clase de hoy



- Tipo de Datos Estructurado
- Clasificación de tipos de Datos Estructurados
- Tipo de datos REGISTRO

CADP – TIPOS DE DATOS

ESTRUCTURADOS



COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

SIMPLE

COMPUESTO

DEFINIDO POR EL LENGUAJE

Integer
Real
Char
Boolean

DEFINIDO POR EL PROGRAMADOR

Subrango

DEFINIDO POR EL LENGUAJE

String

DEFINIDO POR EL PROGRAMADOR

Registros



Supongamos que se quiere representar la información de los distintos inmuebles con los que trabaja una inmobiliaria. Para simplificar el problema la inmobiliaria trabaja con casas o departamentos y sólo los alquila.



Qué información debería conocer la inmobiliaria?

CADP – TIPOS DE DATOS



ESTRUCTURADOS



Tipo propiedad

Cantidad de habitaciones

Cantidad de baños

Precio Alquiler

Localidad

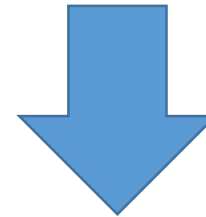
Hasta ahora no conocemos una forma de poder agrupar toda esta información en una sola variable.

CADP – TIPOS DE DATOS

ESTRUCTURADOS



Tipo propiedad
Cantidad de habitaciones
Cantidad de baños
Precio Alquiler
Localidad



Tipo propiedad
Cantidad de habitaciones
Cantidad de baños
Precio Alquiler
Localidad

INMUEBLE



ESTRUCTURA DE DATOS

Permite al programador definir un tipo al que se asocian diferentes datos que tienen valores lógicamente relacionados y asociados bajo un nombre único.

Tipo propiedad
Cantidad de habitaciones
Cantidad de baños
Precio Alquiler
Localidad

INMUEBLE



ESTRUCTURA DE DATOS

Permite al programador definir un tipo al que se asocian diferentes datos que tienen valores lógicamente relacionados y asociados bajo un nombre único.

Elementos

Homogénea

Heterogénea

Acceso

Secuencial

Directo

Tamaño

Dinámica

Estática

Linealidad

Lineal

No Lineal



ELEMENTOS

Depende si los elementos son del mismo tipo o no.

Homogénea



Los elementos que la componen son del mismo tipo

Heterogénea



Los elementos que la componen pueden ser de distinto tipo



TAMAÑO

Hace referencia a si la estructura puede variar su tamaño durante la ejecución del programa.

ESTATICA



El tamaño de la estructura no varía durante la ejecución del programa

DINAMICA



El tamaño de la estructura puede variar durante la ejecución del programa

CADP – TIPOS DE DATOS ESTRUCTURADOS **CLASIFICACION**



ACCESO

Hace referencia a como se pueden acceder a los elementos que la componen.

SECUENCIAL



Para acceder a un elemento particular se debe respetar un orden predeterminado, por ejemplo, pasando por todos los elementos que le preceden, por ese orden.

DIRECTO



Se puede acceder a un elemento particular, directamente, sin necesidad de pasar por los anteriores a él, por ejemplo, referenciando una posición.



LINEALIDAD

Hace referencia a como se encuentran almacenados los elementos que la componen.

LINEAL



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.

NO LINEAL



Para un elemento dado pueden existir 0, 1 ó mas elementos que le suceden y 0, 1 ó mas elementos que le preceden.

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



REGISTRO

Es uno de los tipos de datos estructurados, que permiten agrupar diferentes clases de datos en una estructura única bajo un sólo nombre

Tipo propiedad
Cantidad de habitaciones
Cantidad de baños
Precio Alquiler
Localidad

REGISTRO
INMUEBLE

Una manera natural y lógica de agrupar los datos de cada inmueble en una sola estructura es declarar un tipo **REGISTRO** asociando el conjunto de datos de cada inmueble.

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



Heterogénea



Los elementos pueden ser de distinto tipo (puede haber registros con todos elementos del mismo tipo)

Tipo propiedad
Cantidad de habitaciones
Cantidad de baños
Precio Alquiler
Localidad

Estática



El tamaño no cambia durante la ejecución (se calcula en el momento de compilación)

REGISTRO INMUEBLE

Cómo lo defino?

Campos



Representan cada uno de los datos que forman el registro

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



Program uno;

Const

... .

Type

```
nombre = record  
    campo1: tipo;  
    campo2: tipo;
```

... .

end;

Var

```
variable: nombre;
```



Se nombra cada campo.

Se asigna un tipo a cada campo.

Los tipos de los campos deben ser estáticos.

*Cómo declaro el
registro inmueble?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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;
```

Begin

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



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

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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;
```

Begin

...

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

...

End.

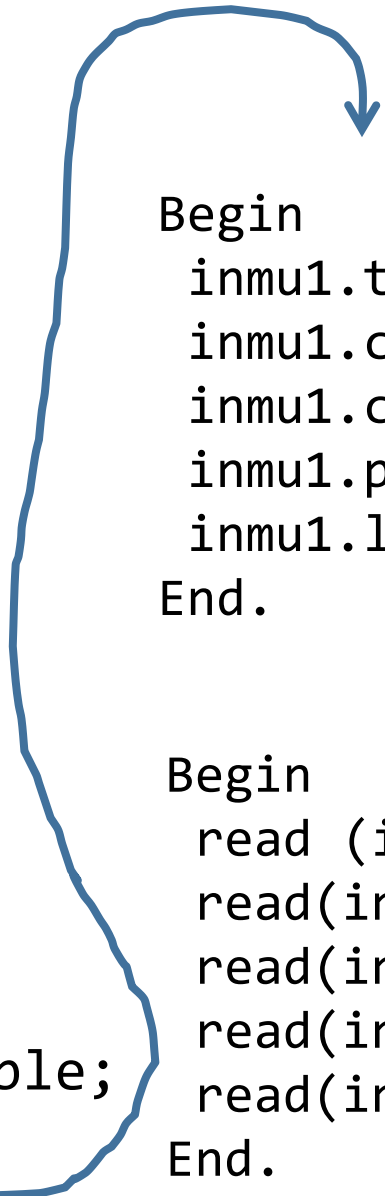


La única forma de acceder
a los campos es
variable.nombrecampo
inmu1.cantHab

CADP – TIPOS DE DATOS ESTRUCTURADOS **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 := 15.500;  
    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?

MODULARIZAR?



No se puede hacer read (inmu1)

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
Procedure leer (var i:inmueble);
```

```
Begin
```

```
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);
```

```
end;
```

Debo asignarlos en el orden en que se declararon? **Qué ocurre si no le doy valor a todos los campos?**

Puede ser una función en vez de un procedimiento?

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 leer (var i:inmueble);
```

```
begin
```

```
...
```

```
end;
```

```
Var
```

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

```
Begin
```

```
    leer (inmu1);
```

```
    inmu2:= inmu1;
```

```
End.
```

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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

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



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

**Qué ocurre si no le
imprimo todos los
campos?**

MODULARIZAR?

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
Procedure imprimir (i:inmueble);
```

```
Begin
```

```
    write(i.tipo);  
    write(i.cantHab);  
    write(i.cantBaños);  
    write(i.precio);  
    write(i.localidad);
```

```
end;
```

**Debo imprimir en el
orden en que se
declararon?**

**Qué ocurre si no
imprimo todos los
campos?**

**Puede ser una
función en vez de un
procedimiento?**

**Cómo comparo
el contenido de
dos registros?**

```
Program uno;
```

```
Const
```

```
...
```

```
Type
```

```
inmueble = record
```

```
    tipo: string;
```

```
    cantHab: integer;
```

```
    cantBaños: integer;
```

```
    precio: real;
```

```
    localidad: string;
```

```
end;
```

```
Procedure imprimir (i:inmueble);
```

```
begin
```

```
...
```

```
end;
```

```
Var
```

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

```
Begin
```

```
    leer (inmu1);
```

```
    imprimir(inmu1);
```

```
End.
```

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
Program uno;
```

```
Const
```

```
...
```

```
Type
```

```
inmueble = record
```

```
  tipo: string;
```

```
  cantHab: integer;
```

```
  cantBaños: integer;
```

```
  precio: real;
```

```
  localidad: string;
```

```
end;
```

```
Var
```

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

```
Begin
```

```
  leer (inmu1);
```

```
  leer (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.
```



No se puede hacer
inmu1 = inmu2

MODULARIZAR?

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
procedure iguales (i1,i2:inmueble;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;  
end;
```

*Puede ser una
función en vez
de un
procedimiento?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
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;
```

```
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;
```

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
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;
```

```
Var
```

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

```
Begin
```

```
    leer (inmu1);
```

```
    leer (inmu2);
```

```
    if (iguales (inmu1,inmu2) = true) then
```

```
        write (`Los registros son iguales`)
```

```
    else write (`Los registros no son iguales`);
```

```
End.
```

```
Begin
```

```
    leer (inmu1);
```

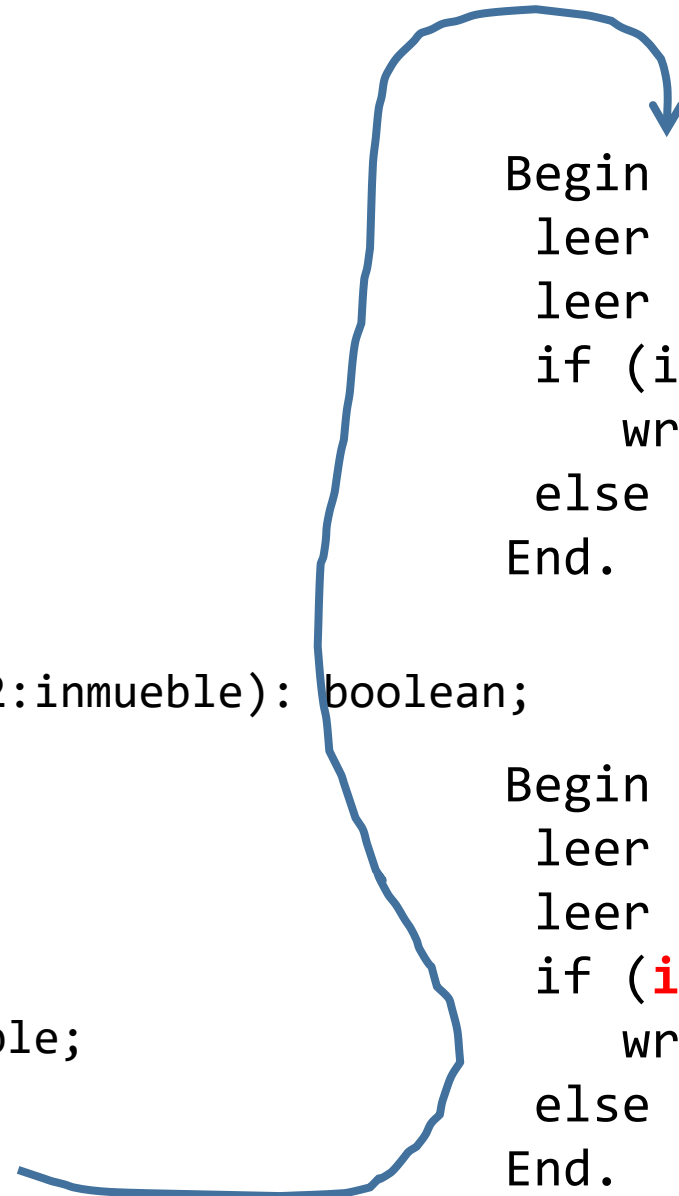
```
    leer (inmu2);
```

```
    if (iguales (inmu1,inmu2)) then
```

```
        write (`Los registros son iguales`)
```

```
    else write (`Los registros no son iguales`);
```

```
End.
```





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



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) then
      if (inmu tiene al menos dos habitaciones) then
        incremento (cant)
      leer registro (inmu)
    end;
  Write (`La cantidad es`, cant);
```

**Como verifico las
condiciones?**

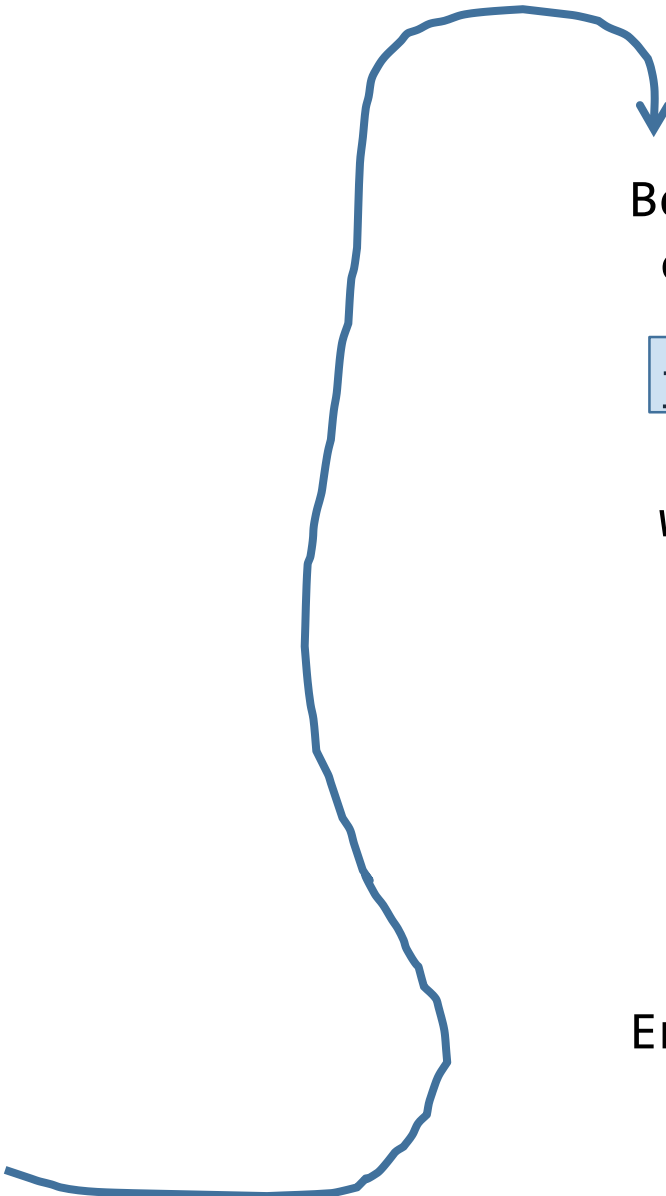
**Qué
modularizo?**

CADP – REGISTROS

EJERCITACION



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



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



```
procedure leer (var i:inmueble);
```

```
Begin
```

```
    read(i.tipo);  
    read(i.cantHab);  
    read(i.cantBaños);  
    read(i.precio);  
    read(i.localidad);
```

```
end;
```

Qué
alternativa
conviene?

```
procedure leer (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;
```



```
function esLaPlata (i:inmueble): boolean;  
var  
    ok:boolean;  
  
begin  
    if (i.localidad = `La Plata`) then  
        ok:= true  
    else  
        ok:= false;  
    esLaPlata:= ok;  
end;
```

Otra opción

```
function esLaPlata (i:inmueble): boolean;  
  
begin  
    esLaPlata:= (i.localidad = `La Plata`);  
end;
```



```
function cantHab (i:inmueble): boolean;  
var  
    ok:boolean;  
  
begin  
    if (i.cantHab >= 2) then  
        ok:= true  
    else  
        ok:= false;  
    cantHab:= ok;  
end;
```

Otra opción

```
function cantHab (i:inmueble): boolean;  
  
begin  
    cantHab:= (i.cantHab >= 2);  
end;
```


CADP – REGISTROS

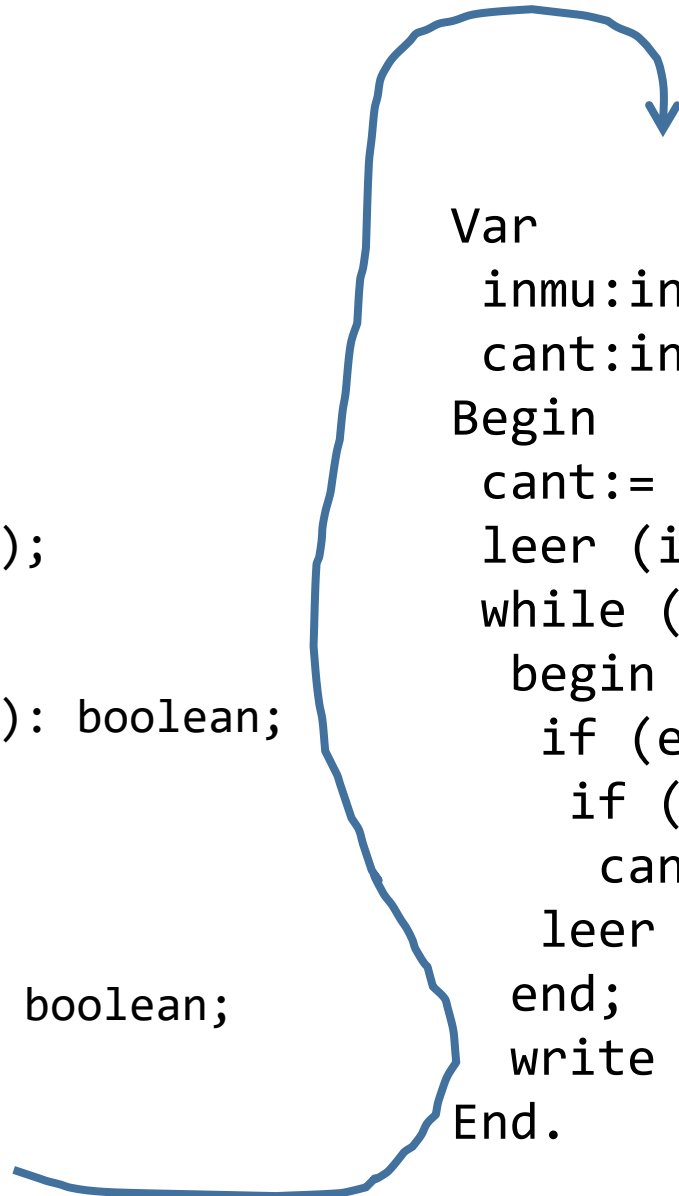
EJERCITACION



```
Program uno;  
Type  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;  
Procedure leer (var i: inmueble);  
begin  
end;  
function esLaPlata (i: inmueble): boolean;  
begin  
    ...  
end;  
  
function cantHab (i: inmueble): boolean;  
begin  
    ...  
end;
```

```
Var  
    inmu: inmueble;  
    cant: integer;  
Begin  
    cant := 0;  
    leer (inmu);  
    while (inmu.localidad <> `XXX`) do  
        begin  
            if (eslapLata (inmu)) then  
                if (cantHab (inmu)) then  
                    cant := cant + 1;  
                leer (inmu);  
            end;  
            write (`La cantidad es`, cant);  
        end;  
    End.
```

*Es necesario pasar
todo el registro a
las funciones?*



CADP – REGISTROS

```
function esLaPlata (i:inmueble): boolean;  
  
begin  
    esLaPlata:= (i.localidad = `La Plata`);  
end;
```

Otra opción

```
function esLaPlata (loc:string): boolean;  
  
begin  
    esLaPlata:= (loc = `La Plata`);  
end;
```

EJERCITACION



```
Begin  
    ...  
    while (inmu.localidad <> `XXX`) do  
        begin  
            if (eslapLata (inmu) then  
                ...  
            end;  
        ..  
    End.
```

```
...  
while (inmu.localidad <> `XXX`) do  
    begin  
        if (eslapLata (inmu.localidad) then  
            ...  
        end;  
    ..  
End.
```

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



Opción 1

```
Program uno;  
Type  
  inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
    dia: integer;  
    mes: integer;  
    año: integer;  
  end;  
Var  
  inmu1: inmueble;
```

*Otra
opción?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



Opción 2

```
Program uno;
```

```
Type
```

```
  fecha = record
```

```
    dia: integer;
```

```
    mes: integer;
```

```
    año: integer;
```

```
  end;
```

```
  inmueble = record
```

```
    tipo: string;
```

```
    cantHab: integer;
```

```
    cantBaños: integer;
```

```
    precio: real;
```

```
    localidad: string;
```

```
    fechaPub: fecha;
```

```
  end;
```

*Cómo hacemos
ahora el proceso
de lectura?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



```
procedure leer (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 leer (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?*

CADP – TIPOS DE DATOS ESTRUCTURADOS **REGISTRO**



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

procedure leer (var i:inmueble);
var
    fec:fecha;
Begin
    read(i.tipo);
    read(i.cantHab);
    read(i.cantBaños);
    read(i.precio);
    read(i.localidad);
    leerFecha (fec);
    i.fechaPub:= fec;
end;
```

Otra opción

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



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` y cuantos inmuebles se publicaron en el verano del 2021.

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

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

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

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

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



3
2



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` y cuantos inmuebles se publicaron en el verano del 2020.

```
Inicializar contadores (cantLP,cantVe)
Leer registro (inmu)
While (no sea el ultimo registro) do
  begin
    if (inmu es de La Plata) then
      incremento (cant)
    if (inmu se publicó en verano 2021) then
      incremento (cantVe)
    leer registro (inmu)
  end;
Write (`Las cantidades son`, cant,cantV);
```

**Qué
modularizo?**

**Como verifico
las condiciones?**

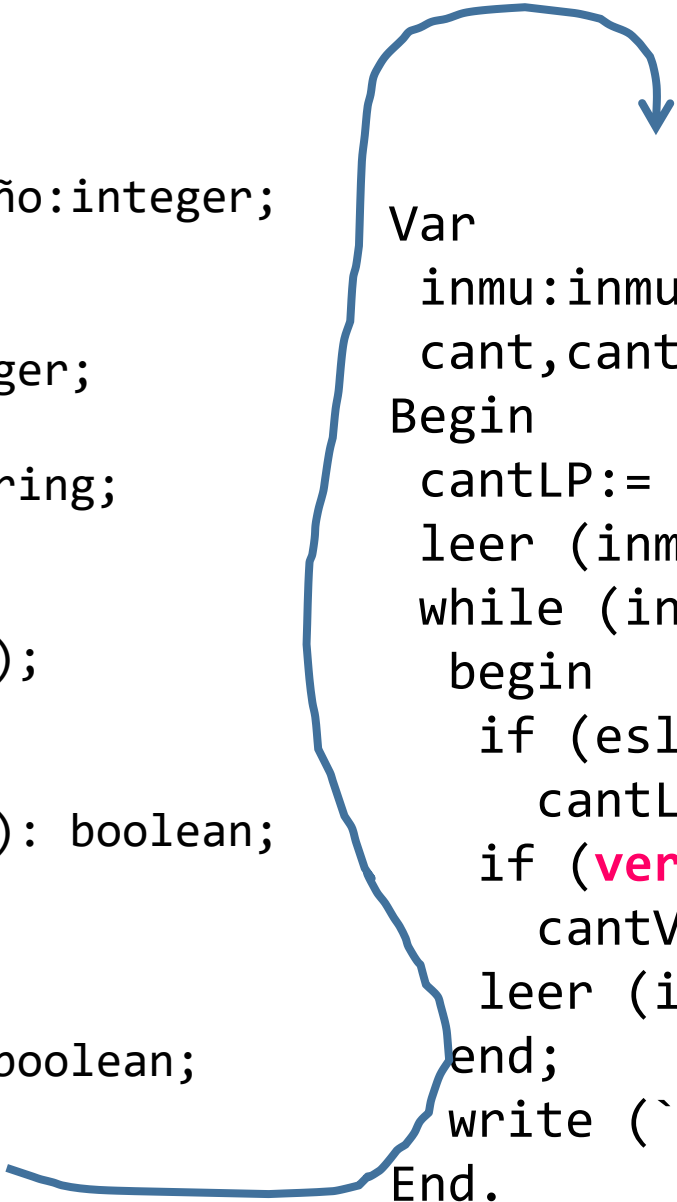
CADP – REGISTROS

EJERCITACION



*Cómo es la
función
verano?*

```
Program uno;
Type
fecha = record
  dia:integer; mes:integer; año:integer;
end;
inmueble = record
  tipo: string; cantHab: integer;
  cantBaños:integer;
  precio: real; localidad: string;
  fechaPub:fecha;
end;
Procedure leer (var i:inmueble);
begin
end;
function esLaPlata (i:inmueble): boolean;
begin
  ...
end;
function verano (i:inmueble): boolean;
begin
  ...
end;
```



```
Var
  inmu:inmueble;
  cant,cantVe:integer;
Begin
  cantLP:= 0; cantVe:=0;
  leer (inmu);
  while (inmu.localidad <> `XXX`) do
  begin
    if (eslapLata (inmu) then
      cantLP:= cantLP + 1;
    if (verano (inmu)) then
      cantVe:= cantVe + 1;
    leer (inmu);
  end;
  write (`La cantidades son`, cantLP,cantV
End.
```

CADP – REGISTROS

EJERCITACION



Otra forma?

```
function verano (i:inmueble): boolean;
```

```
Var
```

```
    cumple:boolean;
```

```
Begin
```

```
    if ( ((i.fechaPub.dia >= 21) and (i.fechaPub.mes = 12))  
        and (i.fechaPub.año = 2020) then cumple:= true
```

```
    else if ( ((i.fechaPub.dia < 21) and (i.fechaPub.mes = 3))  
            and (lug.fechaPub.año = 2021)) then cumple:= true
```

```
    else if ( ((i.fechaPub.mes = 1) or (i.fechaPub.mes = 2))  
            and (lug.fechaPub.año = 2021)) then cumple:= true
```

```
    else cumple:= false;
```

```
verano:= cumple;
```

```
end;
```



Qué cambio y consideraciones debo tener si quiero informar la cantidad de propiedades por localidad?