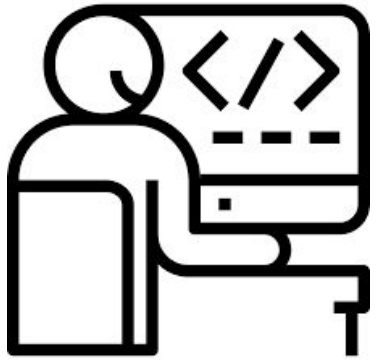


Conceptos de Algoritmos Datos y Programas

CADP – **TEMAS**

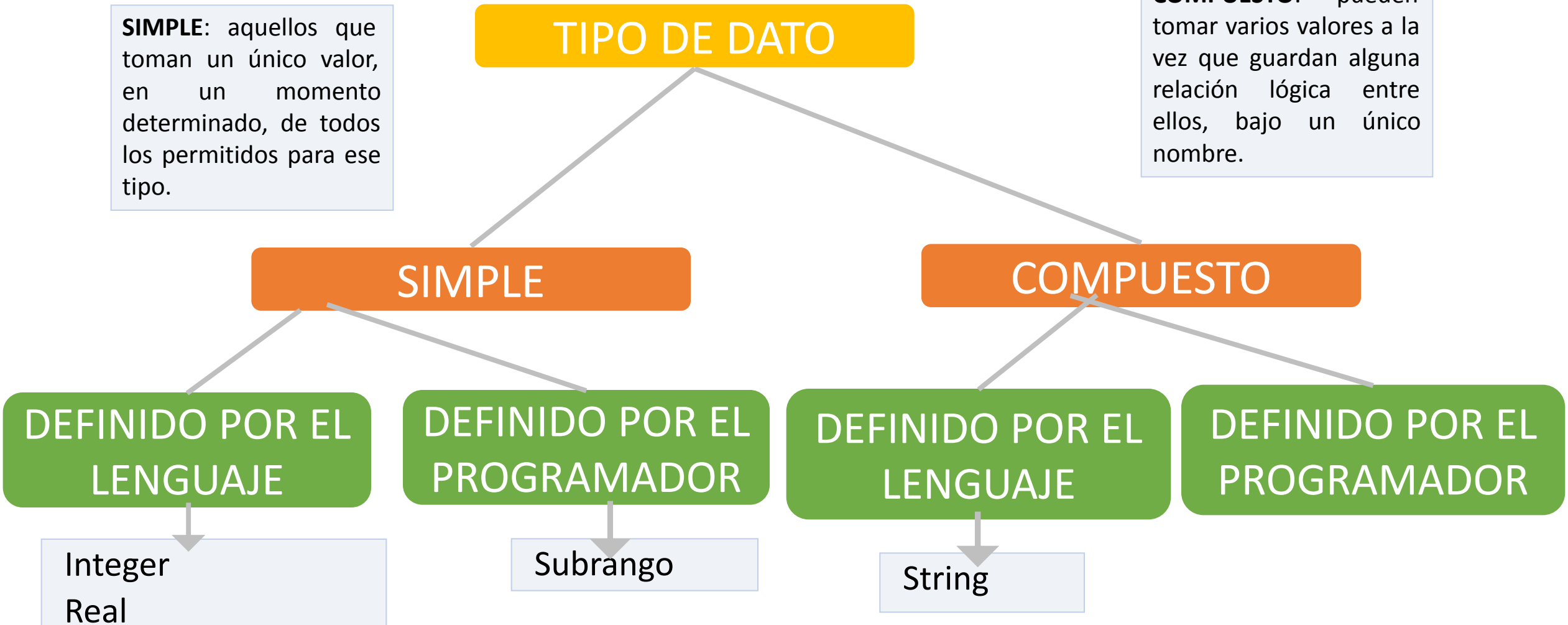
- Estructuras de Datos

CADP – TIPOS DE DATOS

ESTRUCTURADOS

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

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



CADP – TIPOS DE DATOS ESTRUCTURADOS

Supongamos que se quiere representar la información de las distintas razas de animales que existen en una veterinaria. Para simplificar el problema supongamos que la veterinaria atiende solamente perros. De cada animal se conoce la raza, el nombre, la edad.

Qué información es relevante para un perro?

Con lo que sabemos hasta ahora como lo representamos?

CADP – TIPOS DE DATOS ESTRUCTURADOS

Supongamos que se quiere representar la información de las distintas razas de animales que existen en una veterinaria. Para simplificar el problema supongamos que la veterinaria atiende solamente perros. De cada animal se conoce la raza, el nombre, la edad.

Raza
Nombre
Edad

```
Un string para la raza
Un string para el nombre
Un entero para la edad
```

Son variables sueltas pero no hay una sensación de tener la información junta

CADP – TIPOS DE DATOS ESTRUCTURADOS

Supongamos que se quiere representar la información de las 35 materias que forman parte del plan de estudio de una carrera. De cada materia se conoce el nombre, año en que se cursa.

Qué información es relevante para una materia?

```
Un string para el nombre
Un entero para el año
```

Son variables sueltas pero no hay una sensación de tener la información junta

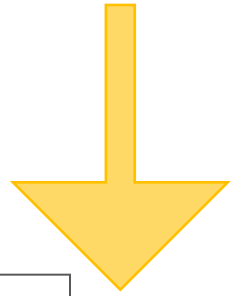
Cómo guardo 35 materias?

CADP – TIPOS DE DATOS ESTRUCTURADOS

Supongamos que se quiere representar la información de los productos que vende un supermercado (la cantidad de productos es variable). De cada producto se conoce el nombre, cantidad que se tiene en stock y precio.

Qué información es relevante para un producto?

```
Un string para el nombre
Un entero para el stock
Un real para el precio
```



Son variables sueltas pero no hay una sensación de tener la información junta

Cómo guardo los productos aun si saber cuántos son?

CADP – TIPOS DE DATOS

ESTRUCTURADOS

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.

CLASIFICACIÓN

Elementos	Acceso	Tamaño	Linealidad
Homogénea	Secuencial	Dinámica	Lineal
Heterogénea	Directo	Estática	No Lineal

CADP – ESTRUCTURA DE DATOS

CLASIFICACION

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

CADP – ESTRUCTURA DE DATOS

CLASIFICACION

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 – ESTRUCTURA DE DATOS

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.

CLASIFICACION

LINEALIDAD

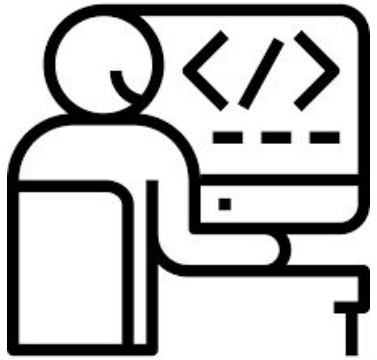
Hace referencia a como se encuentran almacenados los elementos que la componen.

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.



Conceptos de Algoritmos Datos y Programas

CADP – **TEMAS**

- Estructura de Datos REGISTRO

CADP – ESTRUCTURA DE DATOS

REGISTRO

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

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

- TIPO DE DATO
  - SIMPLE
    - DEFINIDO POR EL LENGUAJE
      - Integer
      - Real
      - Char
      - Boolean
    - DEFINIDO POR EL PROGRAMADOR
      - Subrango
  - COMPUESTO
    - DEFINIDO POR EL LENGUAJE
      - String
    - DEFINIDO POR EL PROGRAMADOR
      - Registro

CADP – ESTRUCTURA DE DATOS

REGISTRO

Supongamos que se quiere representar la información de las distintas razas de animales que existen en una veterinaria. Para simplificar el problema supongamos que la veterinaria atiende solamente perros. De cada animal se conoce la raza, el nombre, la edad.

Qué información es relevante para un perro?

Con lo que sabemos hasta ahora como lo representamos?

REGISTRO

Es un tipo de datos estructurado, que permite agrupar diferentes clases de datos en una estructura única bajo un sólo nombre

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

REGISTRO
PERRO

CADP – ESTRUCTURA DE DATOS

REGISTRO

REGISTRO PERRO

Heterogénea

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

Estática

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

Cómo se define?

Campos

Representan cada uno de los datos que forman el registro

CADP – ESTRUCTURA DE DATOS

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 PERRO?

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Const
  ....
Type

  perro = record
   raza: string;
   nombre: string;
   edad: integer;
end;

Var
   ani1, ani2: perro;
```

La característica principal es que un registro permite representar la información en una única estructura.

Cómo se trabaja con un registro?

CADP – ESTRUCTURA DE DATOS

REGISTRO

CON LA VARIABLE REGISTRO

```
Program uno;
Const
   ....
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

Var
    ani1, ani2: perro;
```

```
Begin

 ...
 ani2:= ani1;
 ...
End.
```

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

CADP – ESTRUCTURA DE DATOS

REGISTRO

CON LOS CAMPOS DEL REGISTRO

```
Program uno;
Const
   ....
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

Var
    ani1, ani2: perro;
```

Cómo se le da valor?

```
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

ani1.nombre

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Const
  ....
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

Var
    ani1, ani2: perro;
```

```
Begin
 ani1.raza:='Callejero';
 ani1.nombre:= 'Bob';
 ani1.edad:= 1;
End.
```

```
Begin
 read (ani1.raza);
 read(ani1.nombre);
 read(ani1.edad);
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 (ani1)

CADP – ESTRUCTURA DE DATOS

```
Procedure leer (var p:perro);

Begin
 read (p.raza);
 read(p.nombre);
 read(p.edad);
End.
```

Cómo muestro el contenido de un registro?

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?

REGISTRO

```
Program uno;
Const
  ….
Type
perro = record
   raza: string;
   nombre: string;
   edad: integer;
end;

Procedure leer (var p:perro);
begin
 ….
end;

Var
   ani1, ani2: perro;

Begin
 leer (ani1);
 ani2:= ani1;
End.
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Const
  ….
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

Var
    ani1, ani2: perro;
```

```
Begin
 leer (ani1);
 write (an1.raza);
 write(ani1.nombre);
 write(ani1.edad);
End.
```

Qué ocurre si no le imprimo todos los campos?

No se puede hacer write (ani1)

MODULARIZAR?

CADP – ESTRUCTURA DE DATOS

```
Procedure imprimir (p:perro);

Begin
 write (p.raza);
 write(p.nombre);
 write(p.edad);
End.
```

Cómo se comparan dos registros?

Debo asignarlos en el orden en que se declararon?

Qué ocurre si no le imprimo todos los campos?

Puede ser una función en vez de un procedimiento?

REGISTRO

```
Program uno;
Const
  ....
Type
perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;
Procedure leer (var p:perro);
begin
 ....
end;
Procedure imprimir (p:perro);
begin
 ....
end;
Var
    ani1, ani2: perro;
Begin
 leer (ani1);
 imprimir(ani1);
End.
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Const
   ….
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

Var
    ani1, ani2: perro;
```

```
Begin
 leer (ani1);
 leer (ani2),

 if ((ani1.raza = ani2.raza)and
     (ani1.nombre = ani2.nombre) and
     (ani1.edad = ani2.edad))
  then
   write (`Los registro son iguales`);
End.
```

No se puede hacer
ani1 = ani2

MODULARIZAR?

REGISTRO

```
procedure iguales (p,p1:perro; var ok:boolean);
Begin
    if( (p.raza = p1.raza)and
     (p.nombre = p1.nombre) and
     (p.edad = p1.edad))

    then ok:= true
    else ok:= false;
end;
```

Puede ser una función en vez de un procedimiento?

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
function iguales (p,p1:perro):boolean;
Var
 ok:Boolean;
Begin
    if( (p.raza = p1.raza)and (p.nombre = p1.nombre) and (p.edad = p1.edad))
    then ok:= true
    else ok:= false;
    iguales:= ok;
end;
```

```
function iguales (p,p1:perro):boolean;
Var
 ok:Boolean;
Begin
    ok:= ((p.raza = p1.raza)and
          (p.nombre = p1.nombre)
           and (p.edad = p1.edad))
    iguales:= ok;
end;
```

```
function iguales (p,p1:perro):boolean;
Begin
   iguales := ((p.raza = p1.raza)
                and (p.nombre= p1.nombre)
               and (p.edad = p1.edad));
end;
```

CADP – ESTRUCTURA DE DATOS REGISTRO

```
Program uno;
Const
  ….
Type
perro = record
   raza: string;
   nombre: string;
   edad: integer;
end;
function iguales (p,p1:perro): boolean;
begin
 ….
end;
Procedure leer (var p:perro);
begin
 ….
end;
Procedure imprimir (p:perro);
begin
 ….
end;
```

```
Var
   ani1, ani2: perro;

Begin
 leer (ani1);
 leer (ani2);
 if (iguales (ani1,ani2) = true) then
    write (`Los registros son iguales`)
 else write (`Los registros no son iguales`);
End.
```

```
Begin
 leer (ani1);
 leer (ani2);
 if (iguales (ani1,ani2)) then
    write (`Los registros son iguales`)
 else write (`Los registros no son iguales`);
End.
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

Escriba un programa que lea perros hasta leer un perro cuya raza es `XXX` Al finalizar informe de los perros en con nombre `Bob` y que tienen al menos 2 años

```
Raza `Ovejero`
Nombre `Bob`
edad:2
```

```
Raza `Callejero`
Nombre `Bob`
edad:1
```

```
Raza `Golden`
Nombre `Aragon`
edad:5
```

```
Raza `Ovejero`
Nombre `Lucy`
edad:3
```

```
Raza `Salchicha`
Nombre `Scoby`
edad:1
```

1

CADP – ESTRUCTURA DE DATOS

REGISTRO

Escriba un programa que lea perros hasta leer un perro cuya raza es `XXX` Al finalizar informe de los perros en con nombre `Bob` y que tienen al menos 2 años

```
Inicializar contadores (cant)
Leer registro (ani)
While (no sea el ultimo registro) do
 begin
  if (ani tiene nombre ´Bob´) then
   if (ani tiene al menos dos años) then
    incremento (cant)
  leer registro (ani)
 end;
Write (`La cantidad es`, cant);
```

Cuál es la estructura de datos?

Como verifico las condiciones?

Qué modularizo?

CADP – ESTRUCTURA DE DATOS REGISTRO

```
Program uno;
Const
  ….
Type
perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

// módulos


Var
    ani: perro;
    cant: integer;
```

```
Begin
 cant:=0;

 leer (ani);

 while (ani.raza <> `XXX`) do
  begin
   if (cumpleNombre (ani)  = true) then

     if (edad (ani) = true) then

        cant:= cant + 1;
   leer (ani);
  end;
  write (`La cantidad es`, cant);
End.
```

```
procedure leer (var p:perro);

Begin
   read(p.raza);
   read(p.nombre);
   read(p.edad);
end;
```

Qué alternativa conviene?

```
procedure leer (var p:perro);

Begin
   read(p.raza);
   if (p.raza <> ´XXX´) then
    begin
     read(p.nombre);
     read(p.edad);
    end;
end;
```

```
function cumpleNombre (p:perro): boolean;
var
 ok:boolean;

begin
 if (p.nombre = `Bob`) then
  ok:= true
 else
  ok:= false;
 cumpleNombre:= ok;
end;
```

Otra opción

```
function cumpleNombre (p:perro): boolean;

begin
 cumpleNombre:= (p.nombre = `Bob`);
end;
```

```
function edad (p:perro): boolean;
var
 ok:boolean;

begin
 if (p.edad >= 2) then
  ok:= true
 else
  ok:= false;
 edad:= ok;
end;
```

Otra opción

```
function edad (p:perro): boolean;

begin
 edad:= (p.edad >= 2);
end;
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Type
perro = record
    raza: string;
    edad: integer;
    nombre: string;
end;
Procedure leer (p:perro);
begin
end;
function cumpleNombre (p:perro): boolean;
begin
 ….
end;

function edad (p:perro): boolean;
begin
 ….
end;
```

```
Var
 ani:perro;
 cant:integer;

Begin
 cant:= 0;
 leer (ani);
 while (ani.raza <> `XXX`) do
  begin
   if (cumpleNombre (ani)) then
    if (edad (ani)) then
     cant:= cant + 1;
   leer (ani);
  end;
  write (`La cantidad es`, cant);
End.
```

Es necesario pasar todo el registro a las funciones?

CADP – ESTRUCTURA DE DATOS REGISTRO

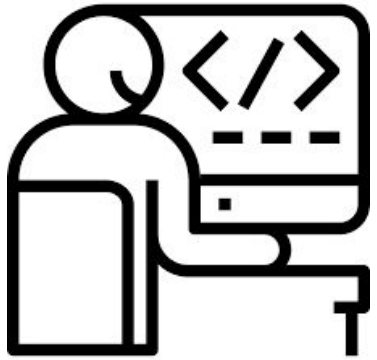
```
Program uno;
Type
perro = record
   raza: string;
   edad: integer;
   nombre: string;
end;
Procedure leer (p:perro);
begin
end;
function cumpleNombre (n:string): boolean;
begin
 ….
end;

function edad (e:integer): boolean;
begin
 ….
end;
```

```
Var
 ani:perro;
 cant:integer;

Begin
 cant:= 0;
 leer (ani);
 while (ani.raza <> `XXX`) do
  begin
   if (cumpleNombre (ani.nombre)) then
    if (edad (ani.edad)) then
     cant:= cant + 1;
   leer (ani);
  end;
  write (`La cantidad es`, cant);
End.
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
function cumpleNombre (nom:string): boolean;
var
 ok:boolean;
begin
 if (nom = ´Bob´) then ok:= true
 else ok:= false;
 cumpleNombre:= ok;
end;

function edad (e:integer): boolean;
var
 ok:boolean;
begin
 if (e>= 2) then ok:= true
 else ok:= false;
 edad:= ok;
end;
```



Conceptos de Algoritmos Datos y Programas

CADP – **TEMAS**

- Estructura de Datos REGISTRO de REGISTRO

CADP – ESTRUCTURA DE DATOS

REGISTRO

Supongamos que se quiere representar la información de las distintas razas de animales que existen en una veterinaria. Para simplificar el problema supongamos que la veterinaria atiende solamente perros. De cada animal se conoce la raza, el nombre, la edad y ahora nuestro perro tiene la fecha en que visitó por última vez la veterinaria

Con se representa ahora?

CADP – ESTRUCTURA DE DATOS

REGISTRO

Opción 1

```
Program uno;
Type
perro = record
    raza: string;
    edad: integer;
    nombre:string;
    dia:integer;
    mes:integer;
    año:integer;
end;

Var
    ani: perro;
```

```
Procedure leer (var p:perro);

Begin
 read (p.raza);
 read(p.nombre);
 read(p.edad);
 read(p.dia);
 read(p.mes);
 read(p.año);
End.
```

Otra opción?

REGISTRO

Opción 2

```
Program uno;
Type
fecha = record
 dia: integer;
 mes:integer;
 año:integer;
end;
perro = record
    raza: string;
    edad: integer;
    nombre: string;
    fechaVis:fecha;
end;
```

Cómo hacemos ahora el proceso de lectura?

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
procedure leer (var p:perro);
Begin
read (p.raza);
 read(p.nombre);
 read(p.edad);
 read(p.fechaVis);
end;
```

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

Como p.fechaVis es otro registro, entonces debo leer campo a campo

```
procedure leer (var p:perro);

Begin
   read (p.raza);
   read(p.nombre);
   read(p.edad);
   read(p.fechaVis.dia);
   read(p.fechaVis.mes);
   read(p.fechaVis.año);
end;
```

Otra alternativa?

REGISTRO

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

```
procedure leer (var p:perro);
var
 fec:fecha;
Begin
   read(p.raza);
   read(p.nombre);
   read(p.edad);
   leerFecha (fec);
   p.fechaVis:= fec;
end;
```

Otra opción

```
procedure leer (var p:perro);
 begin
   read(p.raza);
   read(p.nombre);
   read(p.edad);
   leerFecha (p.fechaVis);
 end;
```

CADP – ESTRUCTURA DE DATOS REGISTRO

Escriba un programa que lea la información de 10 perros Al finalizar informe de los perros que visitaron la veterinaria en los primeros 15 días de los meses de enero y febrero de 2023

Raza `Ovejero`
Nombre `Bob`
edad:2
Dia 8
Mes 1
Año 2022

Raza `Callejero`
Nombre `Bob`
edad:1
Dia 31
Mes 1
Año 2023

Raza `Golden`
Nombre `Aragon`
edad:5
Dia 4
Mes 2
Año 2023

Raza `Callejero`
Nombre `Laly`
edad:4
Dia 7
Mes 1
Año 2023

2

CADP – ESTRUCTURA DE DATOS

REGISTRO

Escriba un programa que lea la información de 10 perros Al finalizar informe de los perros que visitaron la veterinaria en los primeros 15 días de los meses de enero y febrero de 2023

```
Inicializar contador (cant)

Repetir 10
 begin
   Leer registro (ani)
   if (ani cumple fecha) then
    incremento (cant)
 end;

Write (`La cantidad es`, cant);
```

Cuál es la estructura de datos?

Como verifico las condiciones?

Qué modularizo?

CADP – ESTRUCTURA DE DATOS REGISTRO

```
Program uno;
Type
fecha = record
 dia: integer;
 mes:integer;
 año:integer;
end;

perro = record
    raza: string;
    nombre: string;
    edad: integer;
    fechaVis:fecha;
end;
// módulos
Var
    ani: perro;
    cant,i: integer;
```

```
Begin
 cant:=0;

 for i:= 1 to 10 do
  begin

     leer (ani);

       if (cumpleFecha (ani) = true) then
          cant:= cant + 1;
   end;

  write (`La cantidad es`, cant);
End.
```

CADP – ESTRUCTURA DE DATOS REGISTRO

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

procedure leer (var p:perro);
Begin
   read(p.raza);
   read(p.nombre);
   read(p.edad);
   leerFecha(p.fechaVis);
end;
```

```
function cumpleFecha (p:perro): boolean;
var
 ok:boolean;

begin
 if ( ((p.fechaVis.dia >=1) and (p.fechaVis.dia <=15)) and
      ((p.fechaVis.mes =1) or (p.fechaVis.mes =2)) and
      (p.fechaVis.año = 2023) ) then
   ok:= true

 else
   ok:= false;
 cumpleFecha:= ok;
end;
```

CADP – ESTRUCTURA DE DATOS REGISTRO

```
Program uno;
Type
fecha = record
 dia: integer;
 mes:integer;
 año:integer;
end;

perro = record
   raza: string;
   nombre: string;
   edad: integer;
   fechaVis:fecha;
end;


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

procedure leer (var p:perro);
Begin
   read(p.raza);
   read(p.nombre);
   read(p.edad);
   leerFecha(p.fechaVis);
end;
```

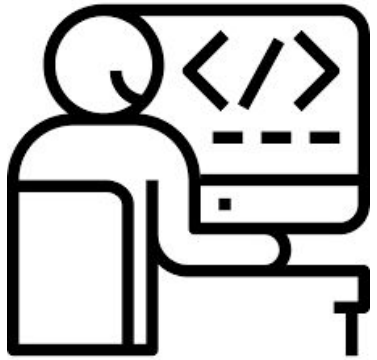
```
function cumpleFecha (p:perro): boolean;
var
 ok:boolean;

begin
 if ( ((p.fechaVis.dia >=1) and (p.fechaVis.dia <=15)) and
      ((p.fechaVis.mes =1) or (p.fechaVis.mes =2)) and
      (p.fechaVis.año = 2023) ) then
   ok:= true

 else
   ok:= false;
 cumpleFecha:= ok;
end;

Var
  ani:perro;
  I,cant:integer;

Begin
 cant:=0;

 for i:= 1 to 10 do
  begin
      leer (ani);
     if (cumpleFecha (ani)  = true) then
      cant:= cant + 1;
   end;

  write (`La cantidad es`, cant);
End.
```

CADP – ESTRUCTURA DE DATOS REGISTRO

```
Program uno;
Type
fecha = record
 dia: integer;
 mes:integer;
 año:integer;
end;

perro = record
    raza: string;
    nombre: string;
    edad: integer;
    fechaVis:fecha;
end;
// módulos
Var
    ani: perro;
    cant,i: integer;
```

```
Begin
 cant:=0;

 for i:= 1 to 10 do
  begin

      leer (ani);

     if (cumpleFecha (ani.fechaVis)=true) then
      cant:= cant + 1;
   end;

  write (`La cantidad es`, cant);
End.
```

```
function cumpleFecha (f:fecha): boolean;
var
 ok:boolean;

begin
 if ( ((f.dia >=1) and (f.dia <=15)) and
      ((f.mes = 1) or (f.mes = 2)) and
      (f.año = 2023) ) then
   ok:= true

 else
   ok:= false;
 cumpleFecha:= ok;
end;
```



Conceptos de Algoritmos Datos y Programas

CADP – **TEMAS**

- Corte de control con REGISTROS

CADP – ESTRUCTURA DE DATOS

REGISTRO

Supongamos que se quiere representar la información de las distintas razas de animales que existen en una veterinaria. Para simplificar el problema supongamos que la veterinaria atiende solamente perros. De cada animal se conoce la raza, el nombre, la edad.
Se pide realizar un programa que lea perros hasta leer uno de raza `XXX` y al final informe la cantidad de perros de cada raza.

Cómo se resuelve?

CADP – ESTRUCTURA DE DATOS

REGISTRO

Se pide realizar un programa que lea perros hasta leer uno de raza `XXX` y al final informe la cantidad de perros de cada raza.

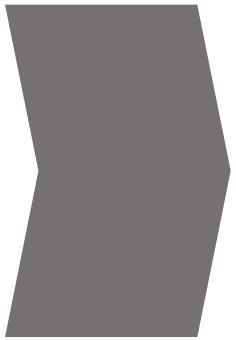
```
Raza `Ovejero`
Nombre `Bob`
edad:2
```

```
Raza `Callejero`
Nombre `Bob`
edad:1
```

```
Raza `Golden`
Nombre `Aragon`
edad:5
```

```
Raza `Callejero`
Nombre `Laly`
edad:4
```

```
Raza `XXX`
Nombre `Laly`
edad:4
```



Ovejero 1

Callejero 2

Golden 1

Qué necesito?

CADP – ESTRUCTURA DE DATOS

REGISTRO

Se pide realizar un programa que lea perros hasta leer uno de raza `XXX` y al final informe la cantidad de perros de cada raza.

```
Raza `Callejero`
Nombre `Bob`
edad:1
```

```
Raza `Callejero`
Nombre `Laly`
edad:4
```

Callejero 2

```
Raza `Golden`
Nombre `Aragon`
edad:5
```

Golden 1

```
Raza `Ovejero`
Nombre `Bob`
edad:2
```

Ovejero1

```
Raza `XXX`
Nombre `Laly`
edad:4
```

NECESITO

Que los datos vengan ORDENADOS

CADP – ESTRUCTURA DE DATOS REGISTRO

Se pide realizar un programa que lea perros hasta leer uno de raza `XXX` y al final informe la cantidad de perros de cada raza. La lectura se encuentra ordenada por raza.

```
Inicializar contador (cant)
Leer registro (ani)
Mientras (no sea la condición de fin) do
 begin
   guardar la raza actual
   mientras (sea la misma raza) do
     begin
      contar perro
      Leer registro (ani)
     end;
   Write (`La cantidad es`, cant);
 end;
```

Cuál es la estructura de datos?

Cómo se que la raza es la misma?

Donde se inicializa el contador?

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

// módulos

Var
    ani: perro;
    cant: integer;
    actual:string;
```

Dónde inicializo cant?

Dónde inicializo actual?

En el write es lo mismo informar actual que ani.raza?

Qué modifico si quiero informar la cantidad total y la por raza?

```
Begin

leer (ani);
while (ani.raza <> ´XXX´) do
 begin
    cant:=0;
    actual:= ani.raza;
    while ((ani.raza <> ´XXX´)and(ani.raza = actual)) do
     begin
          cant:= cant + 1;
          leer (ani);
     end;

     write (`La cantidad de la raza`,actual, ´es´,
cant);
    end;
```

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

// módulos

Var
    ani: perro;
    cant: integer;
    actual:string;
    total:integer;
```

Dónde inicializo cant?

Dónde inicializo actual?

Dónde inicializo total?

```
Begin
total:=0;
leer (ani);
while (ani.raza <> ´XXX´) do
 begin
    cant:=0;
    actual:= ani.raza;
    while ((ani.raza <> ´XXX´)and(ani.raza = actual)) do
     begin
          cant:= cant + 1;
          leer (ani);
     end;
     total:= total + cant;
     write (`La cantidad de la raza`,actual, ´es´,
cant);
 end;
 Write(´La cantidad total de perros es´,total);
```

Qué modifico si quiero informar la raza con mayor cantidad de perros?

CADP – ESTRUCTURA DE DATOS

REGISTRO

```
Program uno;
Type

perro = record
    raza: string;
    nombre: string;
    edad: integer;
end;

// módulos

Var
    ani: perro;
    cant: integer;
    actual:string;
    total:integer;
    max:integer;
    razaMax:string;
```

```
Begin
max:=-1;
leer (ani);
while (ani.raza <> ´XXX´) do
 begin
    cant:=0;
    actual:= ani.raza;
    while ((ani.raza <> ´XXX´)and(ani.raza = actual)) do
     begin
          cant:= cant + 1;
          leer (ani);
     end;
    if (cant >= max) then begin
      max:= cant;
      razaMax:= actual;
    end;
end;
Write (´La raza con más perros es´,razaMax);
End.
```

Dónde inicializo max?

Dónde inicializo cant?