

# PROGRAMACIÓN I

---

TEORÍA – CECILIA SANZ

# Temas

- ✓ Estructura de datos
- ✓ Registro
- ✓ Definición – Operaciones
- ✓ Ejemplos
- ✓ Cláusula WITH

# REPASAMOS

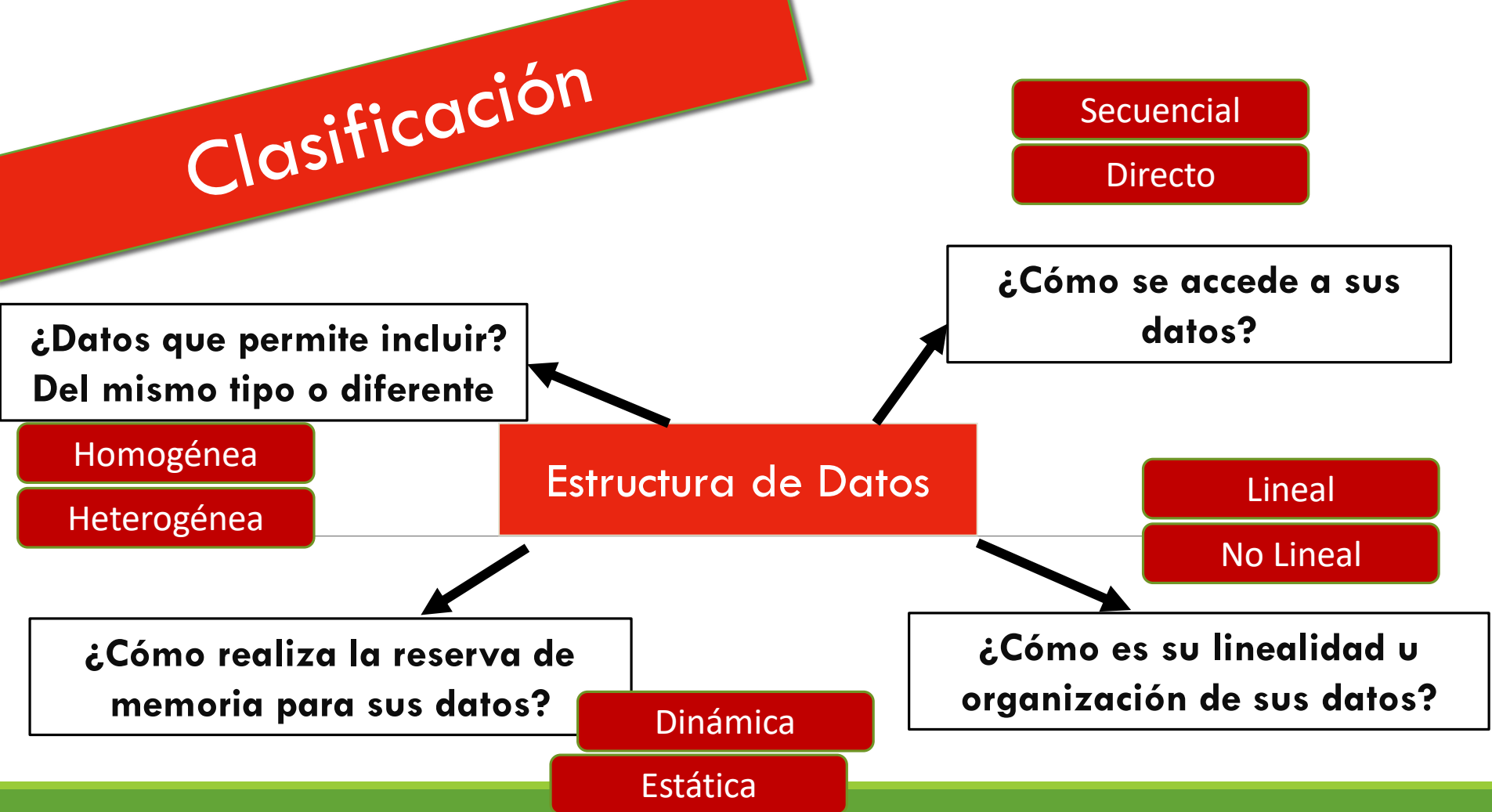


# DATOS Y TIPOS DE DATOS

¿CÓMO REPRESENTAMOS?



# Clasificación





**Motivation**

# Estructura de datos - Ejemplo



**Película**

Título  
Director  
Año



**Persona**

Nombre  
Número de documento  
Domicilio  
Fecha de Nacimiento

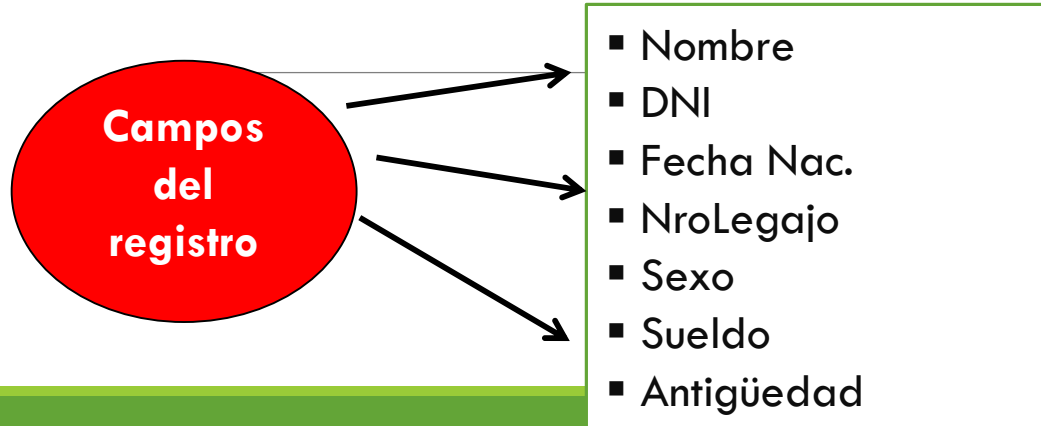
Un **REGISTRO** nos permitirá representar estos datos

# Estructura de datos - Ejemplo

Una manera natural y lógica de agrupar la información de cada empleado en una sola estructura es declarar un tipo REGISTRO asociando el conjunto de datos de cada empleado.

Cada dato que compone el Registro se denomina CAMPO.

## EJEMPLO: REPRESENTAMOS UN EMPLEADO COMO REGISTRO





# REGISTROS

# CLASIFICACIÓN



# Estructura de datos – REGISTRO

## Heterogénea

pelicula

Título: string

Director:string

Año:integer

Categoría: char

## Estática

Título: string

Director:string

Año:integer

Categoría: char

2 string

2 integer

1 char

## Campos

Los valores almacenados en un registro son llamados **campos**, y cada uno de ellos tiene un identificador. Los campos pueden ser nombrados individualmente, como variables ordinarias.

# Estructura de datos – REGISTRO – Ejemplo

Program uno;

**Type** periodo=1950..2014;

cate= A..E;

pelicula = **record**

titulo: string;

director:string;

año:periodo;

categoria: cate;

**end;**

**Var**

pel1, pel2: pelicula;

**Begin**

... **OPERACIONES?**

**End.**

# REGISTROS

# OPERACIONES



## ASIGNACIÓN

# Estructura de datos – REGISTRO

¿Cómo le asignamos valor a un registro?

**Program** uno;

**Type** periodo=1950..2014;

cate= A..E;

pelicula = **record**

titulo: string;

director:string;

año:periodo;

categoria: cate;

**end;**

**Var**

p1:pelicula;

**Begin**

p1.titulo:="La era del hielo";

p1.director:= "Chris Wedge";

p1.año:= 2002;

p1.categoria:='A';

**End.**

¿Qué pasa si no asignamos todos los campos?

# REGISTROS

# OPERACIONES



## ENTRADA Y SALIDA

# Estructura de datos – REGISTRO

¿Cómo le asignamos valor a un registro?

```
Program uno;
```

```
Type
```

```
cadena= string[50];
```

```
codcolor=1..10;
```

```
auto = record
```

```
    marca: cadena;
```

```
    modelo:cadena;
```

```
    precio:real;
```

```
    color: codcolor;
```

```
end;
```

```
Var
```

```
    a1:auto;
```

```
Begin
```

```
    read(a1.marca);
```

```
    read(a1.modelo);
```

```
    read(a1.precio)
```

```
    read(a1.color);
```

```
End.
```

¿Qué pasa si no  
leemos todos los  
campos?

NO SE PUEDE  
read(p1);

¿Cómo  
modularizamos?

# Estructura de datos – REGISTRO

## ¿Cómo le asignamos valor a un registro?

```
Program uno;  
Type  
  cadena= string[50];  
  codcolor=1..10;  
  auto = record  
    marca: cadena;  
    modelo:cadena;  
    precio:real;  
    color: codcolor;  
end;
```

```
Var  a1:auto;  
Procedure Leer (var a: auto);  
Begin  
  readln(a.marca);  
  readln(a.modelo);  
  readln(a.precio);  
  readln(a.color);  
End;  
Begin  
  Leer (a1);  
End.
```

¿Cómo imprimimos  
los datos de un  
registro?



# Estructura de datos – REGISTRO –

## ¿Cómo le imprimimos un registro?

**Program** uno;

**Type**

cadena= string[50];

codcolor=1..10;

auto = **record**

    marca: cadena;

    modelo:cadena;

    precio:real;

    color: codcolor;

**end;**

**Var** a1:auto;

**Procedure** Leer (var a: auto);

**Begin**

    ...

**End;**

**Procedure** Informar (a: auto);

**Begin**

    writeln(a.marca);

**writeln(a.modelo);**

**writeln(a.precio);**

**writeln(a.color);**

**End;**

NO SE PUEDE  
write(a1);

**Begin**

    Leer (a1);

    Informar(a1);

**End.**

# REGISTROS

# OPERACIONES



## COMPARACIÓN

# Estructura de datos – REGISTRO

¿Cómo comparamos dos registros?

Se debe realizar la comparación campo a campo.

NO se puede realizar en forma directa entre dos variables registro, es decir,  
if (a1 = a3) then

# Estructura de datos – REGISTRO

## ¿Cómo comparamos dos registros?

```
Program uno;  
Type  
  auto= record  
    ...  
  end;  
Procedure leer(VAR a:auto);  
begin  
  ...  
end;
```

```
Var a1,a2: auto;  
Begin  
  leer(a1);  
  leer(a2);  
  if (a1.marca = a2.marca)and  
    (a1.modelo = a2.modelo) and  
    (a1.precio = a2.precio) and  
    (a1.color = a2.color)  
  then ....  
End.
```

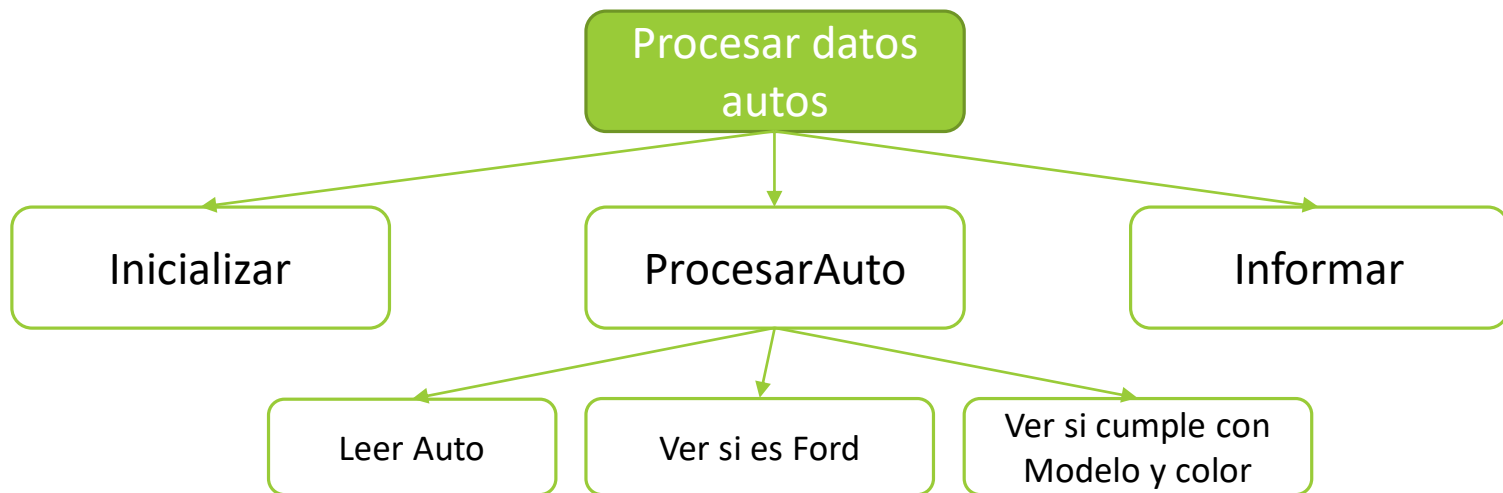
# EJEMPLOS

# EJEMPLOS

# Estructura de datos – REGISTRO – Ejercicio

Se pide realizar un programa que lea datos sobre los autos que ingresan a una agencia de venta, hasta leer uno con marca “XXX”. Al finalizar informar la cantidad de autos de marca “ford”, y la cantidad de autos “ford”, modelo “fiesta” de código de color 5 .

# Diseño de la solución



# Estructura de datos – REGISTRO – Ejercicio

```
Program uno;  
  Const FIN= 'xxx'; MarcaB='Ford';  
  ModeloB='fiesta'; colorB=5;  
  Type codcolor= 1..10;  
    auto= record  
      ...  
    end;  
  Var cantFord,cantTot:integer;  
  Procedure leer(VAR a:auto);  
  begin  
    ...  
  end;  
  Procedure ProcesarAutos(var CantFord,  
    cantTot: integer);  
  Begin  
  End;
```

```
Begin  
  cantFord:=0; cantTot:=0;  
  ProcesarAutos (cantFord, cantTot);  
  Informar(cantFord, cantTot); {Deben implementarlo debajo del leer}  
end.
```



# Estructura de datos – REGISTRO – Ejercicio

```
Procedure ProcesarAutos(var CantFord, cantTot: integer);
Var a: autos;
Begin
  leer (a);
  while (a.marca<>FIN) do
    begin
      if (a.marca=MarcaB) then
        Begin
          cantFord:= cantFord+1;
          if (a.modelo= modeloB) and (a.codcolor=colorB)
            then cantTot:= cantTot+1;
          End;
        leer(a);
      end;
    End;
End;
```

## **Estructura de datos – REGISTRO Ejercicio**

Realice un programa que lea información sobre las app con mayores descargas de google play en el último mes. La lectura se realiza hasta leer una app de nombre “FIN”. De cada app se conoce: nombre, categoría y puntaje. Informe el máximo puntaje dado a una app, y la cantidad de app con categoría “juego”

# Estructura de datos – REGISTRO Ejercicio

Program aplicaciones;

Const fin= “FIN”;

Type

cadena=string[40];

app=**record**

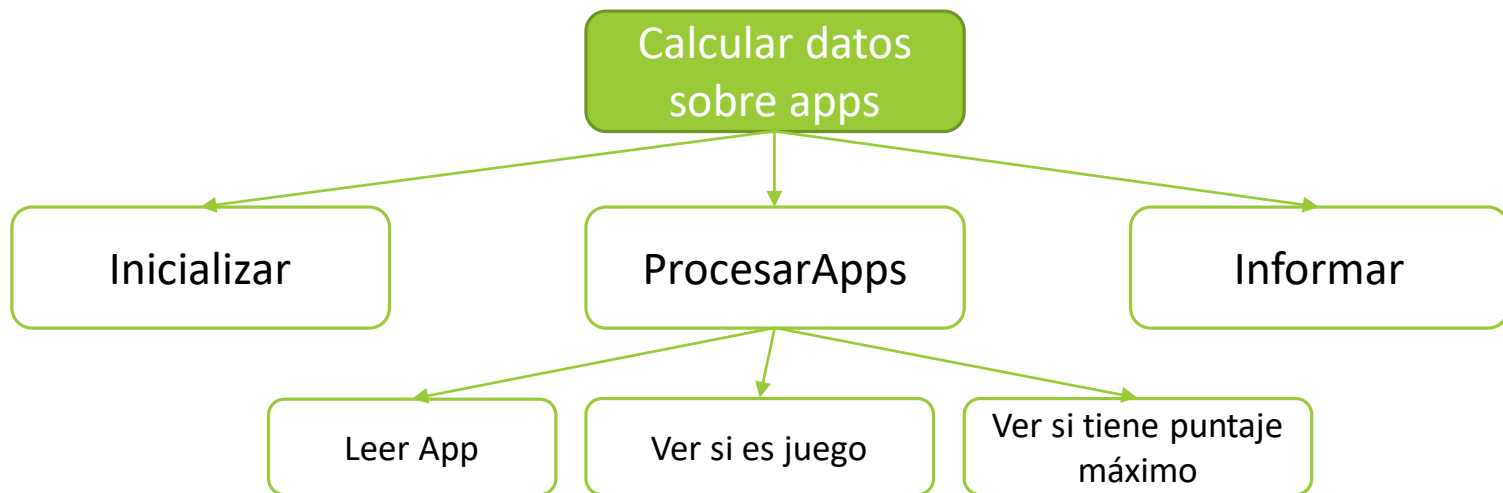
    nombre: cadena;

    categoría: cadena;

    puntaje: real;

**end;**

# Diseño de la solución



# Estructura de datos – REGISTRO Ejercicio

**Var**

cantJuego: integer;

max:real;

maxNom:cadena;

**Begin**

max:=0;

cantJuego:=0; {Inicializar}

ProcesarApp(maxNom, max, cantJuego); {ProcesarApp}

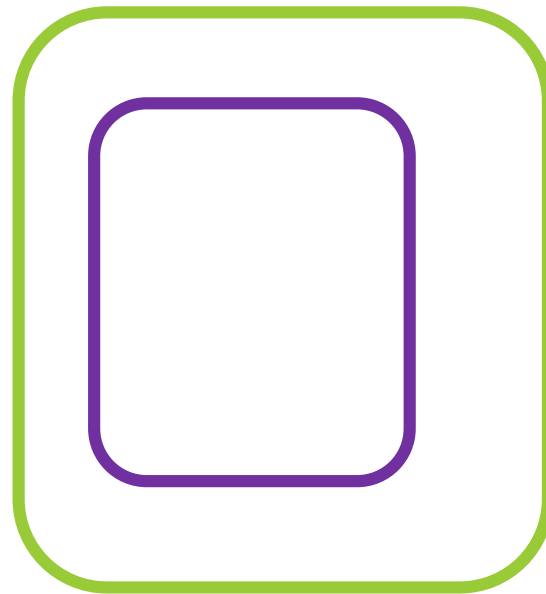
writeln(max, maxNom, cantJuego); {Informar}

**End.**

# Estructura de datos – REGISTRO Ejercicio

```
Procedure ProcesarApp (var maxNom: cadena; var max: real; var cantJuego: integer);  
Var a: app;  
Begin  
  leer(a);                                {Leer}  
  while (a.nombre <> fin) do  
  begin  
    if (a.categoria = 'juego')             {Ver si es juego}  
    then cantJuego := cantJuego + 1;  
    if (a.puntaje > max) then               {Ver si el puntaje es máximo}  
    begin  
      max := a.puntaje;  
      maxNom := a.nombre;  
    end;  
    leer(a);  
  end;  
End;
```

# **REGISTROS ANIDADOS**



# Estructura de datos – REGISTRO de REGISTROS

**Program** anidados;

**type** cadena= string[35]; años:1940..2016;

    dias=1..31; meses=1..12;

    fecha=**record**

        dia:dias;

        mes:meses;

        año:años;

**end;**

    alumno = **record**

        nombre:cadena;

        fNac: fecha;

        añoIngreso:años;

**end;**

**Var** a:alumno;

**Begin**

    a.nombre:="Juan Martini";

    a.fNac.dia:= 2;

    a.fNac.mes:=6;

    a.fNac.año:= 1980;

    a.añoIngreso:= 1999;

**End.**



# Estructura de datos – REGISTRO de REGISTROS

**Program** dos;

**type** cadena= string[35]; años:1940..2016;

dias=1..31; meses=1..12;

fecha=**record**

dia:dias;

mes:meses;

año: años;

**end;**

alumno = **record**

nombre:cadena;

fNac: fecha;

añoIngreso:años;

**end;**

**Var** a:alumno;

**Procedure** leer (**var** a:alumno);

**Procedure** leerFecha(**var** f:fecha);

**begin**

readln(f.dia);

readln(f.mes);

readln(f.año);

**end;**

**Begin**

readln(a.nombre);

leerFecha(a.fNac); {se invoca a su proceso interno}

readln(a.añoIngreso);

**end;**

REGISTROS

CLAUSULA  
WITH

WITH  
WITH  
WITH

# Estructura de datos – REGISTRO – WITH

Cuando se trabaja con registros, hay ocasiones en que el acceso a los campos a través de la calificación suele ser tediosa. Para solucionar este inconveniente, el lenguaje Pascal provee una sentencia **with** que permite que un registro sea nombrado una vez, y luego sea accedido directamente.

# Estructura de datos – REGISTRO – WITH

Su forma general es:

**with** nombre-variable-registro

**do begin**

*{se referencian solo los campos del registro}*

**end;**

¿Cómo lo aplicamos al  
procedimiento leer?

# Estructura de datos – REGISTRO – WITH

---

SIN UTILIZAR WITH

Procedure leer (var alu:alumno)

Begin

    readln(alu.nombre);

    readln(alu.legajo);

    readln(alu.nota);

end;

UTILIZANDO WITH

Procedure leer (var alu:alumno)

Begin

**WITH alu do**

**begin**

            readln(nombre);

            readln(legajo);

            readln(nota);

**end;**

End;

# REGISTROS

Ahora a repasar y  
practicar