



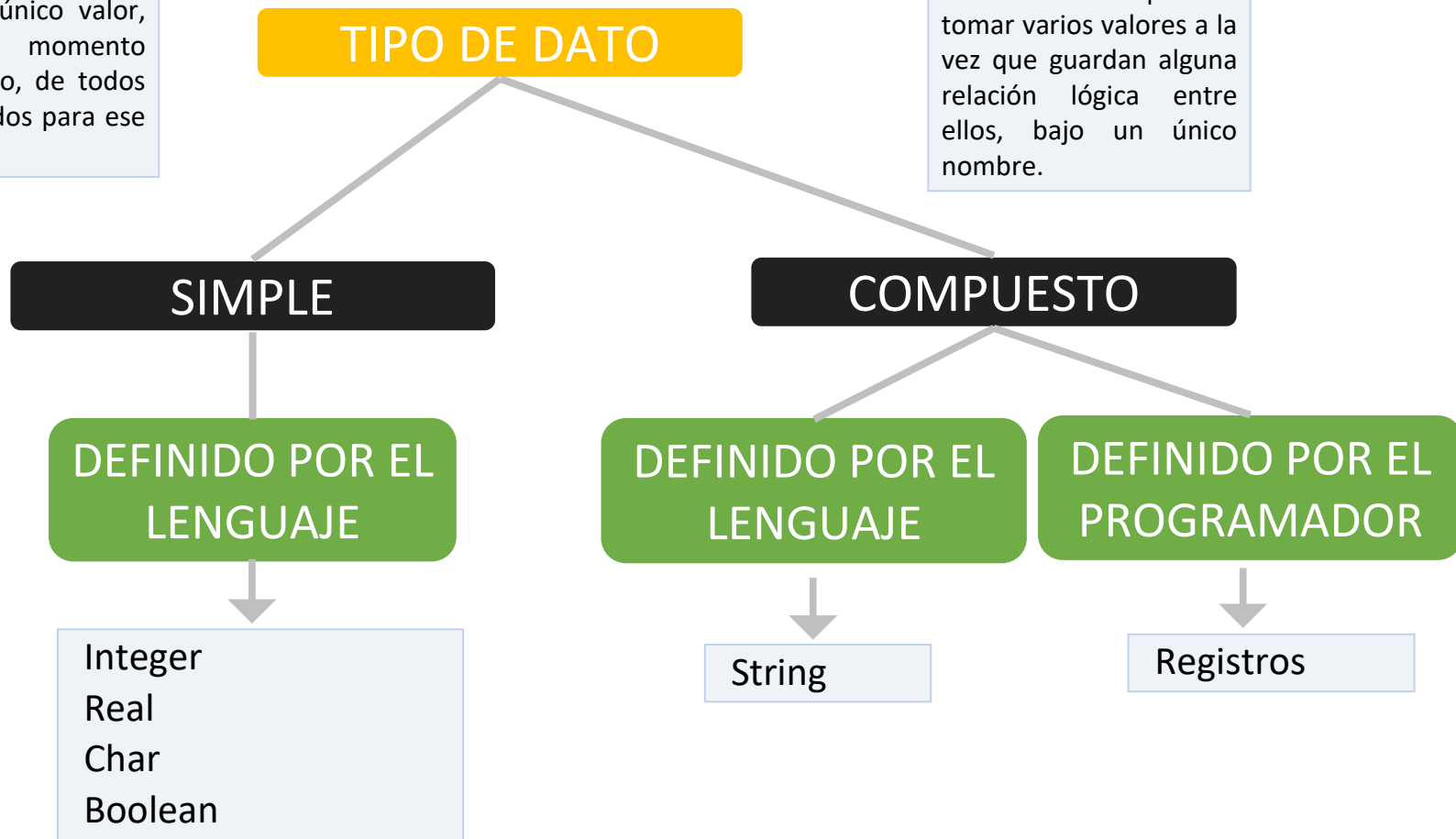
Estructura de Datos Registros - Corte de control

Explicación Práctica 3 (parte 1)

TIPOS DE DATOS

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.



TIPOS DE DATO STRING

- Un tipo de dato string es una sucesión de caracteres de un largo determinado, que se almacenan en un área contigua de la memoria.
- Existe en la mayoría de los lenguajes como un tipo predefinido.
- No es un dato simple.

TIPOS DE DATO STRING

```
Program uno;  
var  
    nombre: string;  
    apellido: string;  
    nombre_y_apellido: string;  
begin  
    nombre:= 'Rosa'; apellido:= 'Gonzalez';  
    nombre_y_apellido:= apellido + ', ' + nombre;  
    write(nombre_y_apellido);  
  
    read(nombre);  
    if(nombre = 'Rosa') then  
        write('Hola Rosita, amiga mia');  
    else  
        write('Hola ' + nombre);  
end.
```



Gonzalez, Rosa

REGISTROS

Definición de tipo

```
tipoRegistro = record  
  campo1: tipo-campo1;  
  campo2: tipo-campo2;  
end;
```

Declaración de variable

```
miVariable: tipoRegistro;
```

Acceso a los datos

```
miVariable.campo1
```

Operaciones

- **Leer un registro:** campo a campo
- **Imprimir un registro:** campo a campo
- **Comparar un registro con otro:** campo a campo
- **Asignar un registro a otro:** usando el operador :=

Ej: $r1 := r2$; ($r1$ y $r2$ son del mismo tipo de registro)

REGISTROS

Ejercicio

Se lee desde teclado una secuencia de sitios turísticos hasta que llegue el sitio con nombre 'fin'. De cada sitio se conoce: nombre, provincia a la que pertenece, cantidad de actividades, cantidad de personas que lo visitaron en un mes.

Se pide:

- a) El nombre del sitio con mayor cantidad de actividades.
- b) La cantidad de sitios con más de 20.000 personas que lo visitaron en un mes.

REGISTROS

A continuación se muestra una lectura de sitios turísticos. Notar cómo se actualizan los valores correspondientes de acuerdo a los datos leídos y analizar por qué en algunas situaciones los valores no se actualizan.

Nombre: Necochea
Provincia: Bs. As.
Cant. Act.: 10
Cant. Visitas: 15.000

Nombre: Ushuaia
Provincia: Tierra del Fuego
Cant. Act.: 9
Cant. Visitas: 21.000

Nombre: Bariloche
Provincia: Rio Negro
Cant. Act.: 20
Cant. Visitas: 20.000

Nombre: San Rafael
Provincia: Mendoza
Cant. Act.: 15
Cant. Visitas: 10.000

Nombre: fin
Provincia:
Cant. Act.:
Cant. Visitas:

Sitio turístico con mayor cantidad de actividades:

Bariloche 20

Cantidad de sitios con más de 20000 visitas:

1

REGISTROS

Implementación

Program turismo;

Type

sitio = record

nombre: string;

prov: string;

cantAct: integer;

cantVis: integer;

end;

Var

sitioTur: sitio;

cant, max: integer;

nomMax: string;

REGISTROS

Implementación

```
{Programa Principal}  
Begin  
    cant:= 0;  
    nomMax:= " ";  
    max:= -1;  
    LeerRegistro(sitioTur);    {Se lee el primer registro}  
    While (sitioTur.nombre <> 'fin') Do begin  
        actualizarMax(sitioTur.cantAct, sitioTur.nombre,  
max, nomMax);  
        if (sitioTur.cantVis > 20000) then  
            cant:= cant + 1;  
            LeerRegistro(sitioTur); {se lee otro registro}  
        end;  
    Write ("Sitio con mas actividades: ", nomMax);  
    Write ("Sitios con más de 20000 visitas: ", cant);  
End.
```

REGISTROS

Implementación

```
Procedure LeerRegistro (var s:sitio);  
begin  
    readln(s.nombre);  
    if (s.nombre <> 'fin') then  
        begin  
            readln(s.prov);  
            readln(s.cantAct);  
            readln(s.cantVis);  
        end;  
end;
```

REGISTROS

Implementación

```
{Programa Principal}  
Begin  
  cant:= 0;  
  nomMax:= " ";  
  max:= -1;  
  LeerRegistro(sitioTur);  {Se lee el primer registro}  
  While (sitioTur.nombre <> 'fin') do begin  
    actualizarMax(sitioTur.cantAct, sitioTur.nombre, max, nomMax);  
    if (sitioTur.cantVis > 20000) then  
      cant:= cant + 1;  
    LeerRegistro(sitioTur); {se lee otro registro}  
  end;  
  Write ("Sitio con mas actividades: ", nomMax);  
  Write ("Sitios con más de 20000 visitas: ", cant);  
End.
```

REGISTROS

Implementación

```
Procedure actualizarMax (cantAct: integer; nombreAct: string; var max:  
    integer; var nommax: string );  
begin  
    if (cantAct > max) then  
        begin  
            max:= cantAct;  
            nommax:= nombreAct;  
        end;  
    end;
```



Estructura de Datos Registros - Corte de control

Explicación Práctica 3 (parte 2)

CORTE DE CONTROL

Ejercicio

Realice un programa que lea de teclado la información de ciudades turísticas (provincia, nombre de ciudad, cantidad de actividades, cantidad de visitantes), ORDENADA POR PROVINCIA, que termina con la provincia 'fin'.

Obtener cuál es la provincia con más visitantes a sus ciudades turísticas.

CORTE DE CONTROL - *Analizando el problema*

Se lee la primera ciudad

La ciudad leída NO pertenece a la prov en proceso

Provincia: Bs. As.
Nombre: Necochea
Cant. Act.: 10
Cant. Visitas: 15.000

Provincia: Bs. As
Nombre: La Plata
Cant. Act.: 9
Cant. Visitas: 21.000

Provincia: **Santa Cruz**
Nombre: El Chaltén
Cant. Act.: 20
Cant. Visitas: 20.000

Provincia: Santa Cruz.
Nombre: El Calafate
Cant. Act.: 15
Cant. Visitas: 35.000

Provincia: **fin**
Nombre:
Cant. Act.:
Cant. Visitas:

Se almacena Bs. As. como la prov que está en proceso. *Se acumula la cant de visitantes para la prov en proceso* *Como se terminó de procesar la prov actual, se actualiza el máximo*

Prov actual

Santa Cruz

Cant. total de
visitantes de la prov
actual

35.000

Prov con mayor
cant de visitantes **Santa Cruz**

cant de visitantes **35.000**

CORTE DE CONTROL - *Implementación*

```
Program ciudades;
```

Type

```
  tciudad = record  
    prov: string;  
    nombre: string;  
    cantAct: integer;  
    cantVis: integer;  
  end;
```

var

```
  ciudadTur: tciudad;  
  max,cantidad: integer;  
  nomMax, provActual: string;
```


CORTE DE CONTROL - Implementación

```
Program ciudades;
```

Type

```
  tciudad = record
```

```
    prov: string;
```

```
    nombre: string;
```

```
    cantAct: integer;
```

```
    cantVis: integer;
```

```
  end;
```

var

```
  ciudadTur: tciudad;
```

```
  max, cantidad: integer;
```

```
  nomMax, provActual: string;
```

*¿ Por qué NO se debe
leer nuevamente
dentro de este while ?*

```
{Programa Principal}
```

```
  nomMax:= ' ';
```

```
  max:= -1;
```

```
  leer(ciudadTur); {Se lee 1er registro}
```

```
  while (ciudadTur.prov <> 'fin') do begin
```

```
    provActual:= ciudadTur.prov;
```

```
    cantidad:=0;
```

```
    while (ciudadTur.prov = provActual) do begin
```

```
      cantidad:= cantidad + ciudadTur.cantVis;
```

```
      leer(ciudadTur); {Se lee otro registro}
```

```
    end;
```

```
    actualizarMax(cantidad, provActual, max,  
nomMax);
```

```
  end;
```

```
  writeln('Prov con mayor cant. de visit. es:',  
nomMax);
```

```
end.
```

*¿ Por qué NO es correcto
enviar ciudadTur.Prov?*

CORTE DE CONTROL - *Implementación*

```
Procedure leer(var c: tciudad);  
begin  
    With c do begin  
        readln(prov);  
        if (prov <> 'fin') then begin  
            readln(nombre);  
            readln(cantAct);  
            readln(cantVis);  
        end;  
    end;  
end;
```

```
Procedure actualizarMax(cantAct: integer; nombreAct: string; var max: i  
    nommax: string);  
Begin  
    if (cantAct > max) then begin  
        max:= cantAct;  
        nommax:= nombreAct;  
    end;  
End;
```