

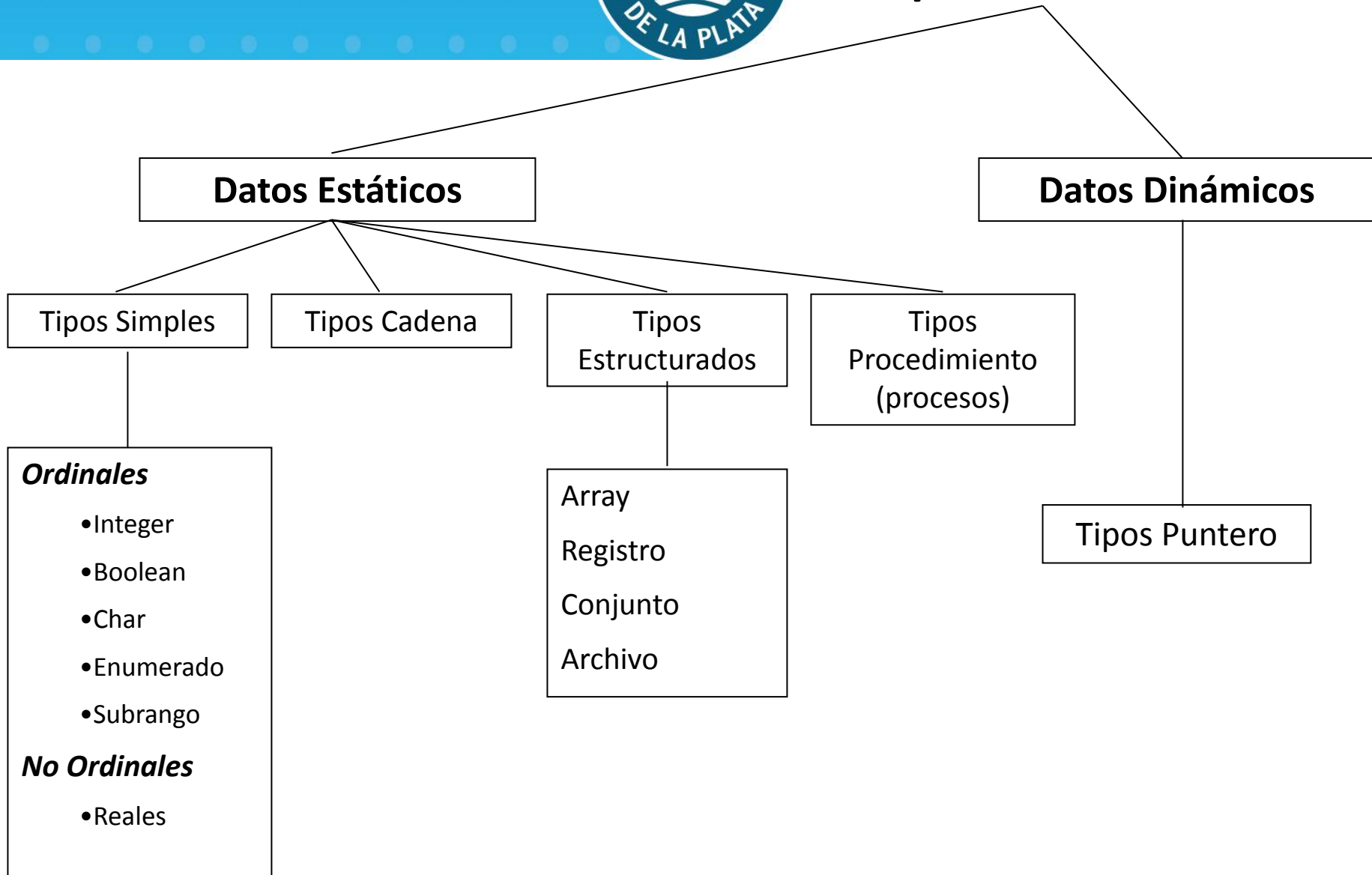


Introducción a la Programación

Estructuras de Datos Estáticas



Tipos de datos





Tipos definidos por el Usuario

- Un tipo de dato define el posible rango de valores que una variable puede tomar al momento de ejecución del programa y a lo largo de toda la vida útil del programa. Este rango está definido por el usuario.
- Tipos de datos definidos por el usuario:
 - Del tipo simple
 - Subrango
 - Enumerados
 - Del tipo estructurado
 - Conjunto



Tipo Subrango

Defición:

Se define a partir de un ordinal estableciendo los límites superiores e inferiores, los subrangos se utilizan para darle una mayor legibilidad a los programas.

```
Type  
  Subrango= lim_inf..lim_sup;  
Var  
  s:Subrango
```

```
Type  
  Meses= 1..12;  
Var  
  mes:Meses;
```

```
Type  
  Dias= 1..7;  
Var  
  dia:Dias;
```



Tipo Enumerado

Definición:

Han de ser del tipo ordinal. Se establecen los valores que van a poder tomar las variables mediante la especificación de una lista de identificadores válidos.

Los identificadores han de comenzar con un carácter y no se pueden repetir. No es posible leer desde teclado ni imprimir directamente sobre la variable.

Type

Dias= (Lunes, Martes, Miercoles, Jueves, Viernes, Sabado, Domingo)

Var

Dia:Dias;

Operaciones y funciones:

De relación: <, >, <>, >=, <=, =

De asignación

Ord (e):

Pred (e):

Succ (e):

Begin

dia:= Jueves;

Ord(dia); // 3

Pred(dia) // Miercoles

Succ(dia) // Viernes

End.



Tipo Conjunto

Un conjunto es una colección de objetos relacionados. Cada elemento dentro del conjunto es llamado miembro o elemento del conjunto.

Aunque en matemáticas no hay restricciones en pascal los objetos:

- El tipo base debe ser un tipo simple, excepto el real.
- Los elementos de un conjunto deben ser del mismo tipo, llamado tipo base.

Type

```
dias_mes = set of 0..31;
```

```
mayusculas = set of 'A'..'Z';
```

```
caracteres = set of char;
```

```
equipos = (Boca, Estudiantes, Racing, Independiente);
```

```
futbol = set of equipos;
```

Var

```
GrupoA, GrupoB : futbol;
```



Tipo Conjunto

Operaciones

Operación	Notación	Conjunto resultante
Unión	+	$A+B$ es el conjunto que contiene todos los elementos que están en A, en B o en ambos.
Intersección	*	$A*B$ es el conjunto cuyos elementos pertenecen a A y B simultáneamente.
Diferencia	-	$A-B$ es el conjunto cuyos elementos son de A pero no de B.

Operadores de relación

Operador	Nombre del operador	Resultado
\leq	Subconjunto	El valor de $A \leq B$ es true. Si cada elemento de A es también de B. En caso contrario es false.
$=$	Igualdad	El valor de $A = B$ es true si cada elemento de A está en B y cada elemento de B está en A. En caso contrario $A = B$ es falso, $A=B$ equivale a $(A \leq B)$ and $(B \leq A)$.
\neq	Desigualdad	El valor de $A \neq B$ es true si el valor de $A = B$ es false y viceversa.
\geq	Superconjunto	$A \neq B$ equivale a $\text{not } (A = B)$, $A \geq B$ es true si $B \leq A$ es true



Clasificación

- Respecto al número de componentes:
 - Estáticas: el número de componentes es fijo durante la ejecución del programa.
 - Dinámicas: el número de componentes varia durante la ejecución del programa.
- Respecto al tipo de los componentes:
 - Por combinación de datos de diferentes tipos (registros).
 - Por repetición de datos del mismo tipo (arrays, conjuntos, archivos, listas, árboles, etc.).



Arreglos (Arrays)

Es una estructura homogénea de datos de tamaño constante. La forma de acceder a cada uno de sus elementos es a través de un identificador común y uno o varios índices.

Características:

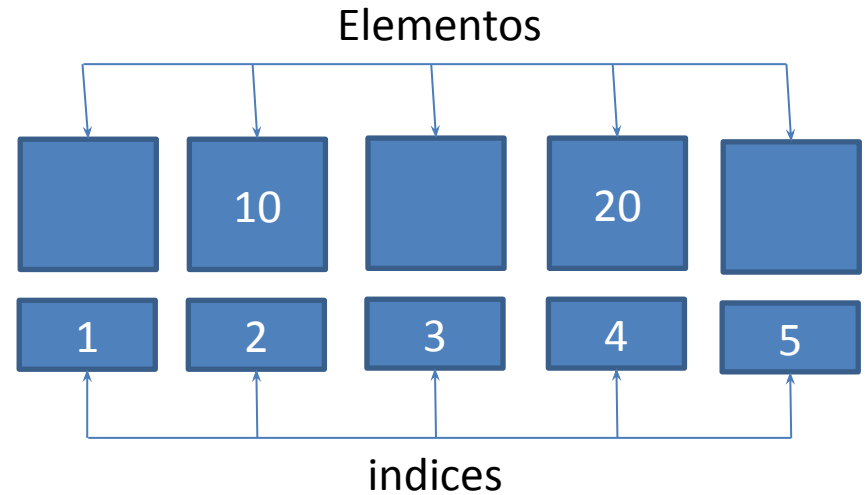
- Todos los elementos del array son del mismo tipo.
- El número de ellos no varía durante la ejecución del programa.
- El número máximo de valores posibles que pueda tomar cada índice se denomina rango de esa dimensión.
- Los valores han de ser consecutivos, por lo que el índice es un ordinal.
- Se accede a la estructura mediante un identificador común (el nombre del array) y un índice.



Arreglos (Arrays) una dimensión

```
Type  
  vector = array[1..5] of integer;  
Var  
  Mivector: vector;
```

```
Mivector[2]:= 10;  
Mivector[4]:= 20;  
  
Longitud:= length (Mivector)  
// Cantidad de elementos del vector (5)
```



Recorrer un vector:

```
For i:=1 to Length (Mivector) do  
  Writeln (Mivector[i]);
```



Arreglos (Arrays) dos dimensiones

```
Type  
  matriz = array[1..4,1..4] of integer;  
Var  
  MiMatriz: matriz;
```

```
MiMatriz[1, 1] := 1;  
MiMatriz[2, 2] := 2;  
MiMatriz[3, 3] := 3;  
MiMatriz[4, 4] := 4;  
MiMatriz[4, 2] := 5;
```

Recorrer una matriz:

```
  fila:=4;  
  columna:=4;  
  for i:=0 to fila do  
    for j:=0 to columna do  
      Writeln (Mimatriz[i, j]);
```

Indices		1	2	3	4
1 2 3 4	1	1			
	2		2		
	3			3	
	4		5		4



Arreglos (Arrays) & Strings

Las cadenas de texto (String) se manejan como si fueran arreglos. Podría decir que son un arreglo de caracteres (char).

```
Type  
  texto = array[1..10] of char;  
Var  
  t: texto;
```

```
T[0]:='H';  
T[1]:='O';  
T[2]:='L';  
T[3]:='A';  
for i:=0 to length(t) do  
  Writeln (t[i]);
```

```
Var  
  t: string;  
Begin  
  t:='HOLA';  
End.
```

```
for i:=0 to length(t) do  
  Writeln (t[i]);
```



Registros

Un registro es una estructura heterogénea de datos, denominados campos y a los que accedemos por nombre. Al igual que cualquier otro dato, el tipo registro (Record) antes de poder ser utilizado debe ser declarado en la sección de tipos.

Type

```
Tiporegistro = record
```

```
    Campo 1 :tipol;
```

```
    ...
```

```
    Campo N : tipoN;
```

```
End;
```

Var

```
    varRegistro: Tiporegistro ;
```

Type

```
TipoPersona = record
```

```
    Nombre: String;
```

```
    Apellido: String;
```

```
    Edad: Integer;
```

```
End;
```

Var

```
    persona1, persona2: TipoPersona;
```

La única operación (aparte de la lectura) que se puede realizar con una variable registro como tal es la asignación, es decir, se pueden copiar todos los campos de una variable registro a otra variable registro del mismo tipo.

```
Persona1.Nombre := "Leo";
```

```
Persona2 := Persona1;
```



Práctica

Antes de codificar, leer atentamente el enunciado!

1. Escriba un programa que almacene en un vector los días de la semana.
2. Escriba un programa que almacene en un vector los números pares del 1 al 10.
3. Escriba un programa que lea las dimensiones de una matriz, visualice la matriz y que encuentre el menor y mayor elemento de esa matriz y sus posiciones.
4. Escribir un programa que en una matriz almacene los números primos del 1 al 100.
5. Dado un arreglo de alumnos con los siguientes datos: legajo, nombre, apellido y 3 notas parciales. Procesar estos datos y construir un nuevo arreglo que contenga solamente los legajos y el promedio de notas de aquellos alumnos cuyo promedio está por encima del promedio general de todos los alumnos.
6. Dado un arreglo de productos (código numérico, descripción) y sus precios de costo. Construya un nuevo arreglo con el código y el precio de venta al público con y sin IVA, sabiendo que debe aumentarle un 50% y el IVA 21%.