

UNIVERSIDAD TECNOLÓGICA NACIONAL

INGENIERIA EN SISTEMAS

Cátedra:

“ALGORITMOS Y ESTRUCTURAS DE DATOS”

UNIDAD Nº 3

DATOS – TIPOS DE DATOS

PROGRAMA EN PASCAL

Ing. Esp. Rosanna M. MAINIERI

UNIDAD 3:

DATOS - TIPOS DE DATOS - PROGRAMA EN PASCAL

- *Datos* 3
- *Tipos de Datos*..... 4
- *Tipo Entero* 5
- *Tipo Real* 6
- *Tipo Alfanumérico*..... 7
- *Tipo Lógico* 8
- *Tipo String* 8
- *Programa en Pascal* 9
- *Bibliografía*..... 13

DATOS

Un programa de computadora opera sobre datos. Un dato es una representación de un objeto del mundo real, mediante la cual se puede modelar aspectos de un problema que se desea resolver.

Puede decirse que:

Dato es un símbolo físico que representa información, tal como un número o un carácter.

Tenemos que representar la información como *datos* antes de poder usarla en la computadora. Estos datos serán procesados estando almacenados internamente en memoria, almacenados en un disco o cinta o introducidos desde una terminal.

Según Dale (1989, p.8):

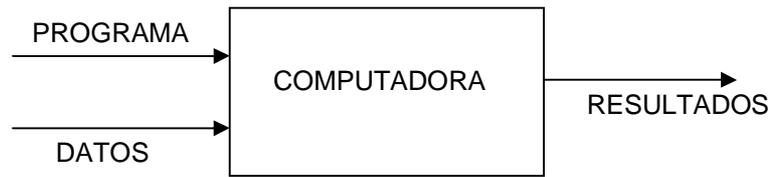
<i>Información</i>	<i>Cualquier conocimiento que puede ser comunicado</i>
<i>Datos</i>	<i>Información que ha sido puesta en forma utilizable por una computadora</i>

En el nivel de hardware y lenguaje de máquina, todos los datos están representados en forma binaria, como secuencia de ceros y unos (bits). Los lenguajes de alto nivel nos permiten emplear abstracciones e ignorar los detalles de representación, principalmente mediante el desarrollo del concepto tipo de dato. Cada tipo de datos se representa de distinta forma en la computadora.

El término Dato desde el punto de vista del Problema

En particular, cuando se desea resolver un problema, se debe hacer una correcta interpretación del enunciado del mismo. Se debe determinar cuáles son los elementos de que se dispone (los *datos* del problema) y cuáles son los objetivos deseados (los resultados que hay que hallar).

Para que la máquina pueda ejecutar las acciones y resolver el problema hay que ingresar el programa y en el momento de ejecución del mismo, se deben ingresar los *datos* con los que trabajará. Aquí estamos haciendo referencia al término *dato* hablando de los objetos que se deben ingresar durante la ejecución del programa. Estos datos serán procesados, se encontrarán resultados intermedios y finalmente se encontrarán los resultados finales deseados.



Desde el punto de vista del problema, diremos que los *datos*, serán los objetos ingresados durante la ejecución del programa.

En cambio, desde el punto de vista de la computadora, todos los objetos procesados serán *datos*, tanto los que se ingresan, como los resultados intermedios y los resultados finales.

	Punto de Vista	Significado
DATO	Problema	Son los objetos (información) que se ingresan durante la ejecución del programa (Datos del problema)
	Autómata	Son todos los objetos que la máquina procesa: <ul style="list-style-type: none"> • Datos de entrada (Datos del problema) • Datos intermedios • Datos de salida (Resultados)

TIPOS DE DATOS

En Pascal (como en la mayoría de los lenguajes) cada elemento de dato debe ser de un tipo específico.

El tipo de dato determina:

- un rango de valores posibles
- cómo se representarán internamente los datos en la computadora, es decir, qué tamaño y forma tendrá la variable donde se almacena.
- qué tipo de procesamiento (operaciones) podrá ejecutar sobre ellos la computadora.

Esto significa que, antes de que podamos escribir cualquier instrucción para manipular datos de un tipo particular, hemos de escribir una declaración que informe de qué tipo es la variable que lo contiene. Toda variable en un programa debe estar asociada a un tipo de dato y sólo a uno.

Un tipo de dato define el conjunto de valores que puede asumir una variable.

Tipo de datos es una familia o conjunto de datos que tienen las mismas propiedades, al que se le asocia un nombre para identificarlo.

O bien

Tipo de datos es la forma general de una clase de elementos de datos.

Hay algunos tipos de datos que se utilizan frecuentemente y el Pascal, como los otros lenguajes, los define automáticamente. Estos son los tipos de datos simples o estándares:

1	Numéricos	Enteros
2		Reales
3	Carácter o Alfanumérico	
4	Lógicos o Booleanos	
5	Cadena o String	

1) Tipo de dato Entero (Integer)

Es el conjunto de los números enteros positivos y negativos (sin parte decimal). Constan de un signo y dígitos.

+22 -16 1 -426 2500

Cuando se omite el signo se asume que el número es positivo.

Teóricamente no hay límite para el tamaño de los enteros, pero dado que la computadora tiene memoria finita, la cantidad de valores que se pueden representar en ella son finitos, por esto se deduce que existe un número entero máximo y otro mínimo.

Se establece una cantidad fija de bytes (conjunto de 8 bits) para almacenar todo dato entero, por ello existe un valor máximo que se pueda representar. Existe un identificador predefinido Maxint, que es el mayor valor entero que pueda representarse.

Pascal dispone generalmente de 2 bytes para almacenar un entero, entonces Maxint es 32767, por lo cual el rango de los enteros permitidos será:

- Maxint hasta Maxint (- 32768 hasta 32767)

Maxint puede ser diferente de una máquina a otra, por lo que puede imprimirse par ver cuál es su valor.

Las expresiones enteras son las que permiten obtener un dato entero. La tabla siguiente muestra los operadores que se pueden utilizar para generar operaciones con enteros. Cinco de ellos dan resultado de tipo entero y uno devuelve un resultado de tipo real.

Operadores			
Operador	Efecto	Tipo de Operando	Tipo de resultado
+	Adición	Entero	Entero
-	Resta	Entero	Entero
*	Producto	Entero	Entero
/	División	Entero	Real
Div	División Entera	Entero	Entero
Mod	Resto de la división entera	Entero	Entero

Las reglas de jerarquía de estos operadores es la misma que la de la aritmética y para igual jerarquía se ejecutan de por orden de aparición. En Pascal no existe la potenciación. Remitirse a la Unidad 2 para ver las funciones que se pueden aplicar a enteros.

2) Tipo de dato Real (Real)

Los datos de tipo real representan a cantidades numéricas reales, es decir, tienen una parte entera y una parte decimal.

Existen dos formas de representarlos:

1. Con parte entera y parte decimal separadas por coma o punto (-348,91)
2. Notación científica (-0.34891 E + 03)

Internamente en la computadora los reales se representan en notación científica con base 2, denominada *coma flotante*, por ser conveniente para la aritmética binaria, definiendo a cada número como una mantisa (parte decimal) y un exponente (posición de la coma).

Se debe tener en cuenta que el tipo de dato real tiene una representación finita de los números reales, dicha representación tiene una precisión fija, es decir, un número fijo de dígitos significativos. Esta condición establece una diferencia con la representación matemática de los números reales. En este caso se tienen infinitos números reales, en tanto que la cantidad de representaciones del tipo de dato real está limitada por el espacio en memoria disponible.

En Pascal se destinan 4 u 8 bytes para representarlos. Dependerá de la cantidad de bytes para tener mayor o menor precisión (cantidad de dígitos significativos) y magnitud (tamaño en valor absoluto) en el número.

Las expresiones reales permiten obtener datos reales. Si en una expresión al menos uno de los operandos es de tipo real, los operadores producen un resultado real.

Operadores	
+	Adición
-	Resta
*	Producto
/	División (ambos operando pueden ser enteros pero el resultado es siempre real)

Para los reales no pueden utilizarse los operadores Div y Mod. Remitirse a la Unidad 2 para ver las funciones que se pueden aplicar a datos de tipo real.

3) Tipo de dato Alfanumérico o Carácter (Char)

Un dato de tipo carácter es **un** carácter simple encerrado entre apóstrofes, este carácter puede ser una letra, un dígito o un símbolo especial.

Por ejemplo: 'A' 'a' '8' '-' '\$' '' '?'

Cada máquina tiene un conjunto de caracteres alfanuméricos que pueden ser representados en ella. Este conjunto se llama el conjunto de caracteres de la máquina e incluye:

- Letras 'A'...'Z', 'a'...'z'
- Dígitos '0'...'9'
- Caracteres especiales '&', '+', '_', ' ', '?'

Existen tablas donde aparecen los distintos caracteres. La más conocida es la Tabla ASCII (American Standard Code for Information Interchange) y es la que usa Pascal. Esta tabla presenta un conjunto de 256 caracteres, numerados de 0 a 255, es decir que, cada carácter tiene un número de orden, por ende, es un conjunto ordenado. Por ser un conjunto ordenado se pueden aplicar los operadores relacionales y las funciones para conjuntos ordenados.

Obsérvese que cada carácter está encerrado entre apóstrofes o comillas simples. El compilador necesita los apóstrofes para diferenciar entre el dato carácter '8' ó '+' y el entero 8 ó el operador suma +.

Las siguientes funciones se pueden aplicar sobre los datos de tipo carácter:

ORD (X)	Devuelve el número de orden, en la Tabla ASCII, del carácter X
CHR (N)	Devuelve el carácter correspondiente al número de orden N, de la Tabla ASCII

Se advierte que ORD y CHR son funciones inversas, es decir:

$$\text{CHR}(\text{ORD}(C)) = C \quad \text{y} \quad \text{ORD}(\text{CHR}(T)) = T$$

SUCC (X)	Devuelve el carácter siguiente al X en la Tabla ASCII
PRED (X)	Devuelve el carácter anterior al X en la Tabla ASCII

4) Tipo de dato Lógico o Boolean (Boolean)

Booleano es un tipo con sólo dos valores: True y False (Verdadero y Falso).

Los valores true y false representan un conjunto ordenado, donde false precede a true, es decir: false < true.

Combinando operandos con operadores lógicos se obtienen expresiones lógicas o booleanas.

Los datos booleanos no pueden ingresarse como datos (como los otros tres tipos estándares), pero sí pueden imprimirse.

Remitirse a la tabla de la Unidad 2 para observar las funciones que producen resultados lógicos.

5) Tipo de dato Cadena (String)

El tipo de dato String es una sucesión de caracteres que se almacenan en un área contigua de la memoria y que puede leído o escrito.

Este tipo represente un dato de tamaño dado que resulta de la concatenación (unión) de los caracteres que lo forman. Al declarar la variable de tipo string se debe indicar la longitud que es el número máximo de caracteres que puede contener.

```
Var Nombre_Variable_string : string [ longitud]
```

Ejemplo:

```
CADENA: string [10];
```

```
...
```

```
CADENA := 'lunes' ;
```

Las operaciones válidas sobre string son la asignación y la concatenación, ésta permite adosar un string a continuación de otro.

Los operadores relacionales pueden utilizarse con los string (=, <, <=, >, >=, <>), no es necesario que las dos variables string tengan la misma longitud.

PROGRAMA EN PASCAL

Un programa consiste en una cabecera (presentación del programa) y dos partes esenciales:

1. una descripción de las instrucciones (acciones) que deben ser ejecutadas.
2. una descripción de los datos que son manipulados por esas acciones.

Las acciones se describen mediante las llamadas *sentencias* y los datos mediante *declaraciones* y *definiciones*.

El lenguaje Pascal, como los otros lenguajes, tienen palabras reservadas que habrá que utilizar para escribir correctamente un programa.

Palabras reservadas: Una palabra que tiene un significado especial en el lenguaje Pascal (o cualquier otro lenguaje) y no puede ser utilizada como identificador.

Partes de un programa

Los programas en Pascal, entonces, se componen de:

Cabecera
Sección de Declaraciones y Definiciones (D. y D.)
Sentencias

Cabecera

Indica el comienzo del programa, se le dá un nombre (identificador) definido por el usuario y se establecen los parámetros a través de los cuales el programa se comunica con su entorno.

Estos parámetros pueden ser los dispositivos de entrada/salida de la computadora o archivos. Input hace referencia al dispositivo estándar de entrada (teclado) y Output hace referencia al dispositivo estándar de salida (monitor).

Program Ejemplo (Input, Output);

Sección de Declaraciones y Definiciones (D. y D)

En esta sección se definen todos los objetos locales al programa. Consta de las siguientes cinco secciones:

1. Declaración de Rótulos
2. Definición de Constantes
3. Definición de Tipos
4. Declaración de Variables
5. Declaración de Procedimientos y Funciones

1. Declaración de Rótulos (Label)

Se puede marcar cualquier sentencia de un programa anteponiéndole un rótulo. De este modo se puede referir a la misma mediante la sentencia GO TO. En esta sección, encabezada por la palabra Label, se declaran los rótulo que luego serán utilizados.

Si bien, el lenguaje Pascal acepta la rotulación de sentencias para el uso de la sentencia GO TO, el paradigma de Programación Estructurada con el que nosotros trabajamos, no lo permite (Ver Unidad N° 1).

2. Definición de Constantes (Const)

Una definición de constante sirve para introducir un identificador como sinónimo de un dato constante. Este valor constante puede ser un número o un carácter. Esta sección comienza con la palabra Const

El empleo de constantes mejora la legibilidad del programa y actúa como una ayuda de documentación conveniente además de facilitar la futura modificación del programa.

Ej. Const CANT=40;

3. Definición de Tipos (Type)

El lenguaje Pascal permite al usuario (programador) que defina sus propios tipos de datos para un programa, además de los tipos de datos estándar.

La palabra Type encabeza esta sección. Aquí se define el nuevo tipo y se le asocia un nombre.

4. Declaración de Variables (Var)

Toda variable que aparezca en alguna sentencia debe estar declarada en una *declaración de variable*. Esta declaración debe preceder a cualquier uso que se haga de la variable.

Una declaración asocia un identificador (nombre) y un tipo de dato con una nueva variable. Se lista el identificador seguido de su tipo. La palabra Var encabeza esta sección.

Ej. Var A: Integer;
 SUELDO: Real; LETRA: Char

Esta asociación identificador-tipo es válida para todo el bloque que contiene la declaración, excepto que el identificador sea redefinido en un bloque subordinado (Tema que se tratará más adelante)

5. Declaración de Procedimientos y Funciones (Procedure y Function)

Todo proceso o subprograma, ya sea Procedimiento o Función, debe definirse (o anunciarse) antes de ser utilizado. En esta sección es donde se los declara. La sección comienza con la palabra Procedure para declarar a un Procedimiento y comienza con la palabra Function para declarar a una Función. Se formalizará el tema en unidades más adelante.

Sentencias

La sección de sentencias es la parte ejecutable del programa y está encerrada por un par Begin – End.

Las instrucciones de esta sección son las que se ejecutan luego de la compilación.

Veamos un ejemplo de un programa en Pascal:

```
Program VENTA (Input, Output);

(* La primera línea de este programa es la cabecera, las siguientes son las
declaraciones*)

Const       PORC = 12;       (* Porcentaje de descuento*)
Var         IMPORTE, DESCUENTO, IMPFINAL: Real;

(* Las sentencias ejecutables comienzan aquí *)

Begin
  Read (IMPORTE);
  DESCUENTO:= IMPORTE * PORC / 100;
  IMPFINAL:= IMPORTE – DESCUENTO;   (* Importe final a pagar *)
  Write ('El importe final es ', IMPFINAL)
End.
```

En el programa aparecen comentarios que describen al mismo. Los comentarios pueden aparecer en cualquier sitio del programa y están delimitados por (* *) ó { }. El compilador ignora lo escrito como comentario, se utilizan sólo para facilitar la lectura de los programas.

Observar que las sentencias están separadas por un punto y coma (excepto delante del End, en donde no es necesario). Los puntos y comas (;) son separadores, no son parte de una sentencia. Un programa siempre termina con un punto (.). Los comentarios que van después del punto son ignorados.

BILBIOGRAFÍA

DALE Nell, WEEMS Chip, (1989), *PASCAL*, Ed. Mc. Graw Hill, España

DE GIUSTI Armando, (2001), *Algoritmos, Datos y programas*, Ed. Prentice Hall, Fac. de Informática, Univ. Nac. La Plata, Argentina.

JENSEN Kathleen, WIRTH Niklaus, (1987), *PASCAL Manual del usuario e informe*, Ed. El Ateneo, Argentina.

JOYANES AGUILAR Luis, (1996), *Fundamentos de Programación*, Ed. Mc Graw-Hill, España.