Conceptos de Algoritmos, Datos y Programas
2018

TEORIA 1

TEMAS de la CLASE

1. Concepto Informática
2. Etapas de resolución de un problema por computadora
3. Concepto de Algoritmo
4. Concepto de Programa
5. Concepto de Datos y Tipos de Datos
6. Clasificación de Tipos de Datos Simples definidos por el lenguaje (Entero y Real)
7. Declaración de Datos en Pascal
8. Estructura general de un programa en Pascal
9. Ejercitación

Concepto de Informática

La Informática es la ciencia que estudia el análisis y resolución de problemas utilizando computadoras.

Ciencia se relaciona con una metodología fundamentada y racional para el estudio y resolución de los problemas.

La ***resolución de problemas*** aplicaciones en áreas muy diferentes tales como biología, comercio, control industrial, administración, robótica, educación, arquitectura, etc.

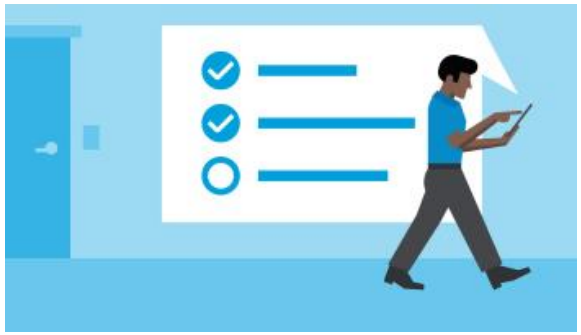
Computadora máquina digital y sincrónica, con capacidad de cálculo numérico y lógico y de comunicación con el mundo exterior. Ayuda al hombre a realizar tareas repetitivas en menor tiempo y con mayor exactitud.

Etapas de resolución de un problema por computadora
Del Problema a la Solución

Problema del Mundo Real

Análisis

Modelo



1er etapa: se sintetizan los requerimientos del problema y se simplifica el contexto y los datos a utilizar por el programa en la computadora.

Etapas de resolución de un problema por computadora
Del Problema a la Solución

Problema del Mundo Real

Análisis

Diseño

Algoritmos

Solución Modularizada

2da etapa: la descomposición funcional nos ayudará a reducir la complejidad, a distribuir el trabajo y en el futuro a re-utilizar los módulos.

Algoritmos.

Etapas de resolución de un problema por computadora
Del Problema a la Solución

Problema del Mundo Real

Análisis

Diseño

Implementación

Programa

3er etapa: escribir algoritmos en un lenguaje de programación y elegir la representación de los datos.

Etapas de resolución de un problema por computadora
Del Problema a la Solución

Problema del Mundo Real

Análisis

Diseño

Implementación

Verificación

4ta etapa: verificar que el programa conduce al resultado deseado, utilizando datos representativos del problema real

¿Qué puede ocurrir?

Etapa de Análisis del Problema

- En una primera etapa, se analiza el problema en su contexto del mundo real. Deben obtenerse los **requerimientos** del usuario. El resultado del análisis del problema es un modelo preciso del ambiente del problema y del objetivo a resolver.

Etapa de Diseño de la solución

- Suponiendo que el problema es computable, a partir del modelo se debe diseñar una solución. En el paradigma procedural, esta etapa involucra entre otras tareas la modularización del problema, considerando la descomposición del mismo y los datos necesarios para cumplir su objetivo.
- Esta etapa involucra la especificación de los algoritmos:
- Cada uno de los módulos del sistema diseñado tiene una función que podemos traducir en un algoritmo (que puede no ser único). La elección del algoritmo adecuado para la función del módulo es muy importante para la eficiencia posterior del sistema de software.

Etapa de Implementación de la solución

- Esta etapa involucra la escritura de los programas. Los algoritmos definidos en la etapa de diseño se convierten en programas escritos en un lenguaje de programación concreto.

Etapa de Verificación de la solución

- Una vez que se tienen los programas escritos y depurados de errores de sintaxis, se debe verificar que su ejecución conduce al resultado deseado, utilizando datos representativos del problema real.

En resumen:

Análisis y Diseño

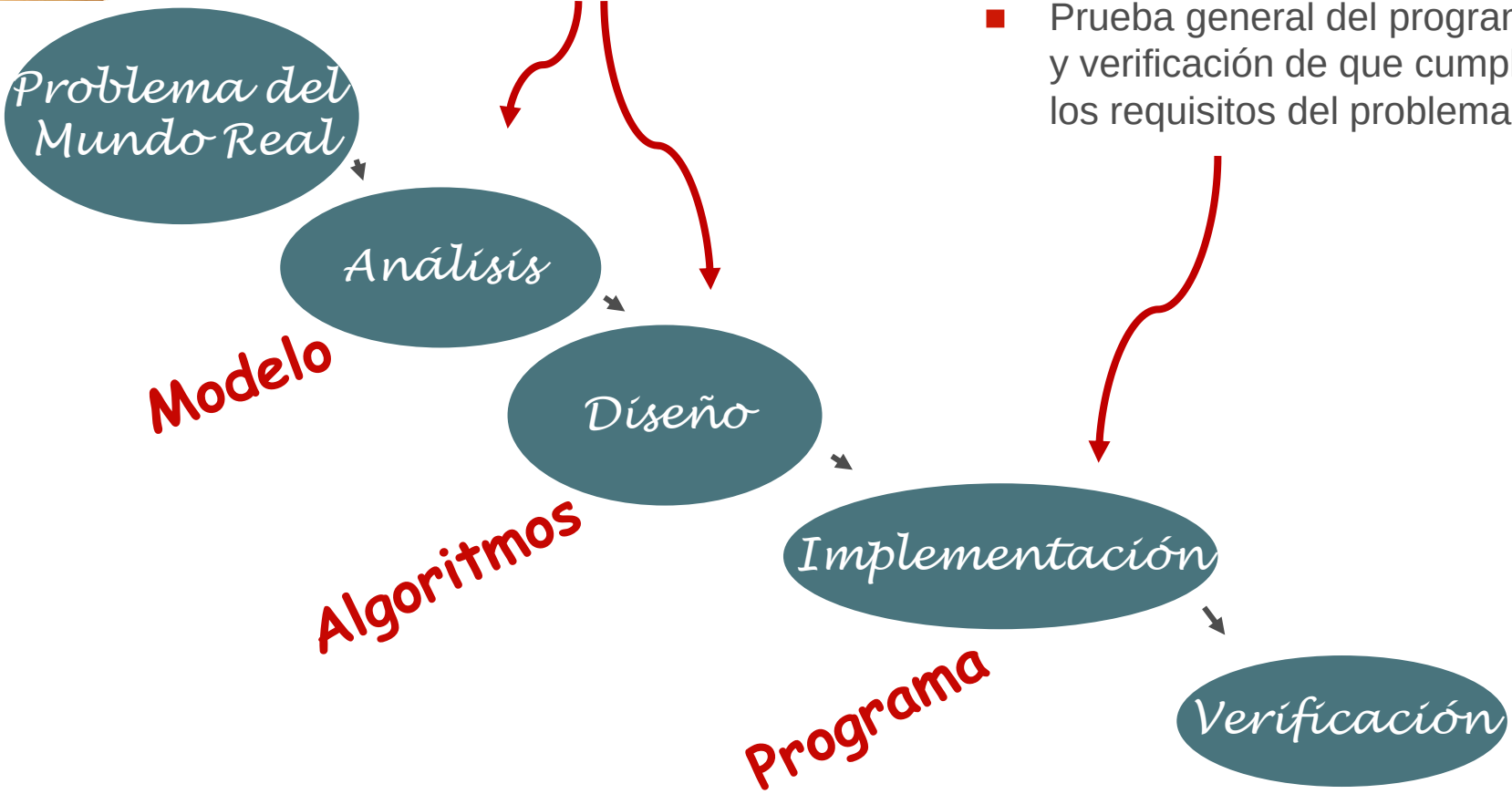
(*NO depende del lenguaje*)

- Entender el problema
- Modelizarlo.
- Modularizar.
- Escribir los algoritmos.

Implementación

(*depende del lenguaje*)

- Codificación de los algoritmos en un lenguaje de programación.
- Prueba en ejecución de cada módulo.
- Prueba general del programa y verificación de que cumple los requisitos del problema.



Concepto de Algoritmo

Definiremos **algoritmo** como la **especificación rigurosa** de la secuencia de pasos (instrucciones) a realizar sobre un **autómata** para alcanzar un resultado deseado en un **tiempo finito**.

✓***Especificación rigurosa*** significa que debemos expresar un algoritmo en forma clara y unívoca.

✓Alcanzar el resultado en ***tiempo finito*** significa que suponemos que un algoritmo ***comienza y termina***. *Por lo tanto*, el número de instrucciones debe ser también finito.

✓ Si el autómata es una computadora, tendremos que escribir el algoritmo en un lenguaje "entendible" y ejecutable por la máquina.

PROGRAMA

Concepto de Programa

Definiremos **programa** como el conjunto de instrucciones u órdenes **ejecutables** sobre una **computadora**, que permite cumplir con una función específica (dichas órdenes están expresadas en un lenguaje de programación concreto).

- ✓Normalmente los programas alcanzan su función objetivo en un **tiempo finito**.
- ✓ Los **programas de aplicación** constituyen el verdadero valor que da utilidad a las computadoras (programas WEB, de administración, cálculo, comunicaciones, control industrial, sistemas expertos etc.).
- ✓ Los **programas** se escriben en un **lenguaje de programación** siguiendo un conjunto de reglas sintácticas y semánticas.

Escribir un programa exige:

✓***Elegir la representación*** adecuada de los datos del problema**.**

✓***Elegir el lenguaje de programación*** a utilizar, según el problema y la máquina a emplear.

✓***Definir el conjunto de instrucciones*** (en el lenguaje elegido) cuya ejecución ordenada conduce a la solución.

Programa = Instrucciones + Datos

Programación Imperativa

El modelo que siguen los lenguajes de programación para **DEFINIR** y **OPERAR** la información, permite asociarlos a un paradigma de programación particular.

En esta primera parte del curso trabajaremos bajo el paradigma imperativo/procedural.

Lenguaje de programación: PASCAL

Programa = Instrucciones + Datos

Las ***instrucciones*** (***acciones)*** representan las operaciones que ejecutará la computadora al interpretar el programa.

Se escriben en un lenguaje de programación determinado

PASCAL

Los ***datos*** son los valores de información de los que se necesita disponer y en ocasiones transformar para ejecutar la función del programa.

Cada lenguaje de programación tiene los propios

Concepto de Dato

Un **Dato** es una representación de un objeto del mundo real. Los datos permiten modelizar los aspectos del problema que se quieren resolver mediante un programa ejecutable en una computadora.

Datos constantes: datos que no cambian durante la ejecución del programa.

Datos variables: datos que durante la ejecución del programa pueden cambiar.

En PASCAL cada dato debe tener asociado un tipo de dato

Definición de Tipo de Dato

Un *tipo de dato* es una clase de objetos de datos ligados a un conjunto de operaciones para crearlos y manipularlos.

Los tipos de datos se caracterizan por:

- ✓ Un rango de valores posibles
- ✓ Un conjunto de operaciones realizables sobre ese tipo
- ✓ Una representación interna

¿Para que sirven?

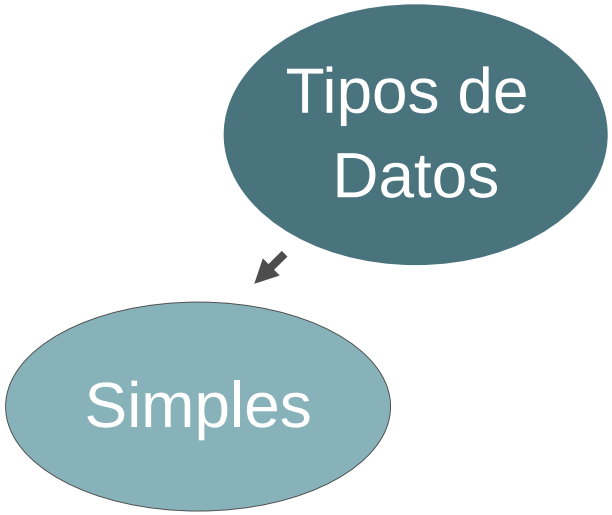
¿Qué valores puedo usar?

¿Qué operaciones puedo aplicar?

¿Cuánta memoria utiliza?

Clasificación de Tipos de Datos

Una primera clasificación



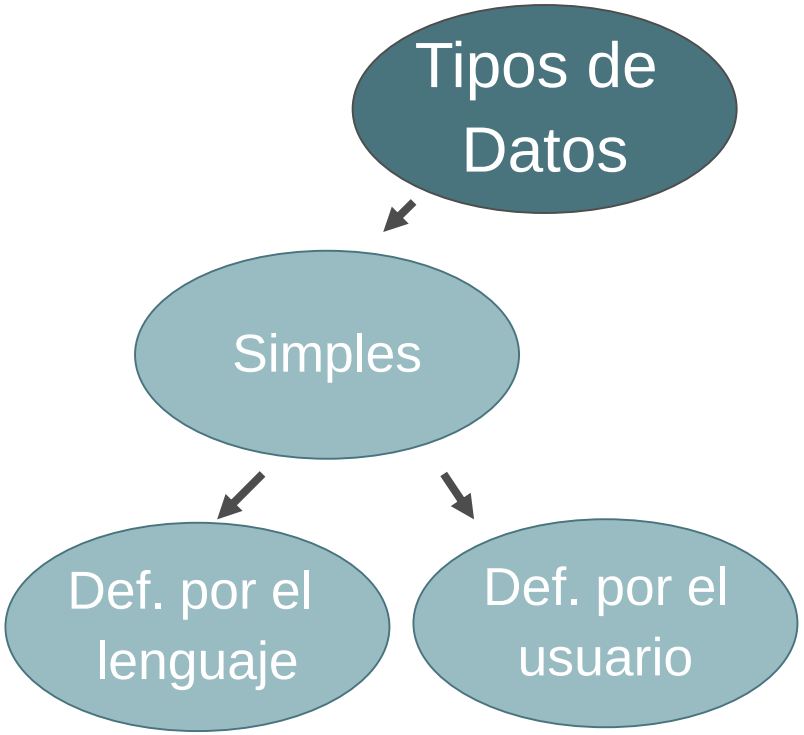
Los tipos de datos simples son aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

Clasificación de Tipos de Datos Simples

A su vez, los tipos simples, pueden clasificarse en:

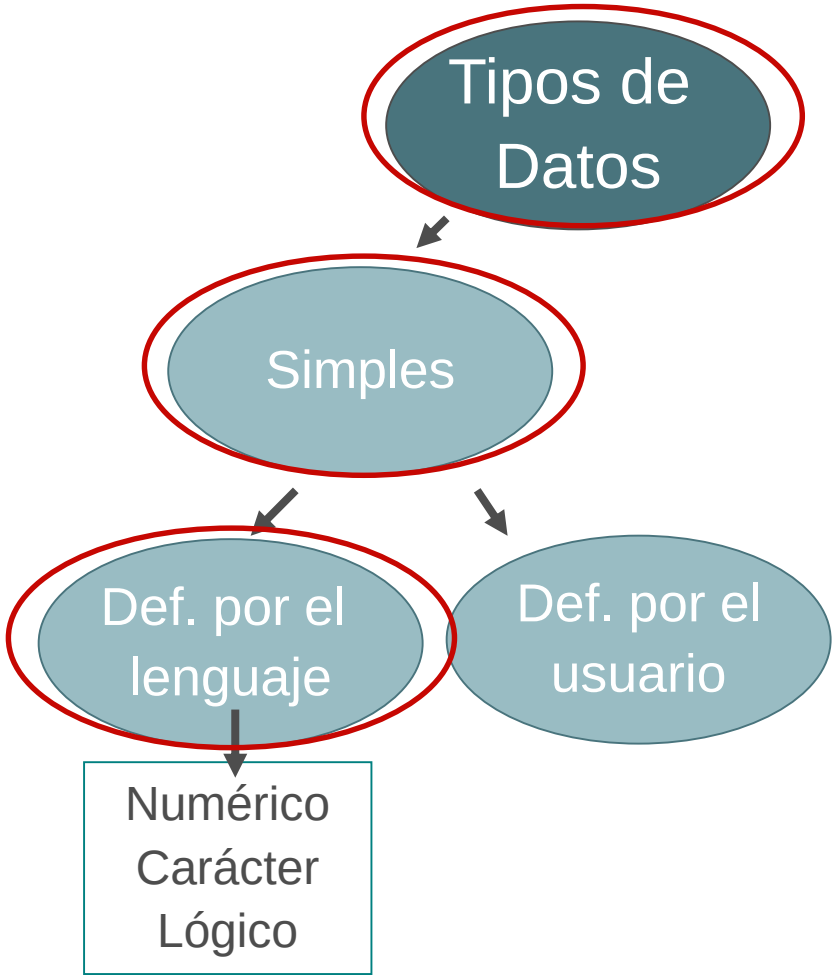
- Los tipos de datos definidos por el lenguaje (primitivos o estándar) son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

- Los tipos definidos por el usuario, permiten definir nuevos tipos de datos a partir de los tipos primitivos.



Clasificación de Tipos de Datos Simples Definidos por el lenguaje

Comenzaremos presentando los tipos simples y definidos por el lenguaje



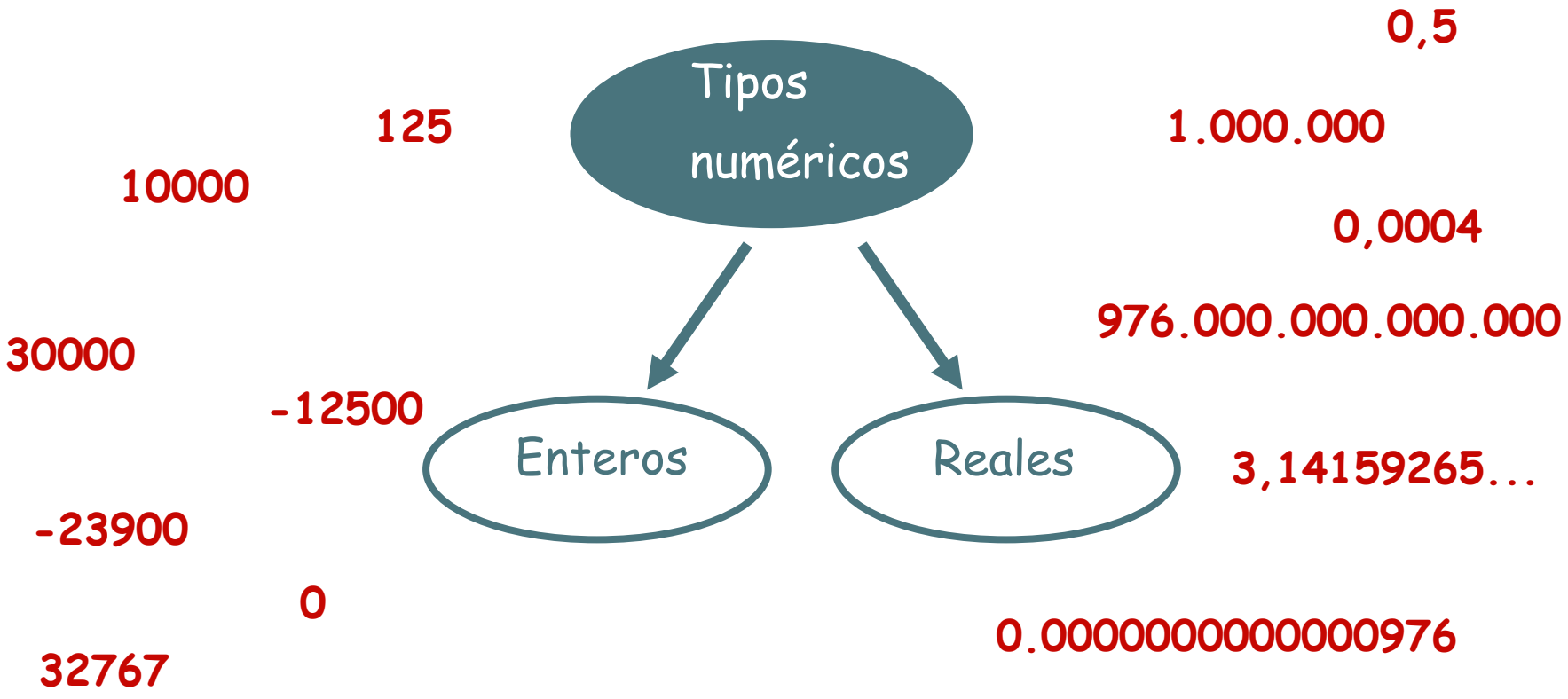
¿Valores posibles?

¿Operaciones?

¿Representación en memoria?

Tipo de dato numérico

El **tipo de dato numérico** permite representar el conjunto de los valores numéricos que puede referirse a los números enteros o a los números reales:



Tipo de Dato numérico: ENTERO

El tipo de dato entero es un tipo de dato simple y ordinal.

Los elementos son del tipo: ..., -3, -2, -1, 0, 1, 2, 3, ...

Dado que una computadora tiene memoria finita, la cantidad de valores enteros que se pueden representar sobre ella son finitos, por esto se deduce que existe un número entero máximo y otro mínimo.

Hay sistemas de representación numérica que utilizan 16 dígitos binarios (bits) para almacenar en memoria cada número entero, permitiendo un rango de valores enteros entre -2^{15} y $+2^{15}$. Entonces podrán representarse los enteros entre -32768 y +32767

La ocupación en memoria es de 2 bytes

Tipo de Dato numérico: REAL

El tipo de dato real es una clase de dato numérico que permite representar números decimales. Es un tipo de dato simple.

✓ El tipo de dato real tiene una representación finita de los números reales.

✓ Se utiliza la notación exponencial o científica, utilizada por cualquier tipo de calculadora, y consiste en definir cada número como una *mantisa* (parte decimal) y un *exponente* (posición de la coma).

976.000.000.000.000 -> 9,76 x 10^{14}

0.000000000000976 -> 9,76 x 10^{-14}

La ocupación en memoria es de 6 bytes

Operaciones del tipo de dato numérico

Operaciones aritméticas

Las operaciones válidas para el tipo de dato numérico son:

- ✓**suma** (+),
- ✓**resta** (-),
- ✓**multiplicación** (*),
- ✓**división** (/),
- ✓**división entera** (div)
- ✓**módulo** (mod).

Operador	**Tipo de Operando**	**Tipo de resultado**
Suma (**+**)	Entero ó Real	Entero ó Real
Resta (**-**)	Entero ó Real	Entero ó Real
Multiplicación (*****)	Entero ó Real	Entero ó Real
División (/)	Real	Real
Div	Entero	Entero
Mod	Entero	Entero

3+4

2,5 / 3

10 div 4

10 mod 4

Orden de precedencia de las operaciones con números

Las expresiones que tienen dos o más operandos requieren reglas matemáticas que permitan determinar el orden de las operaciones.

El orden de precedencia para la resolución de la expresión matemática es:

En primer lugar: ***** y **/**

Luego: **+** y **-**

Por último: **div** y **mod**

- 6 + 8 * 5 = 6 + 40 = 46
- (6 + 8) * 5 = 14 * 5 = 70
- 5 * 2 + 7 + 4 * 3 = 10 + 7 + 12 = 17 + 12 = 29
- 5 * (2 + 7 + 4) * 3 = 5 * 13 * 3 = 195
- 5 + 10 mod 4 = 3
- 10 div 2 * 3 = 1
- (5 * 4) div 4 = 5

Se pueden utilizar paréntesis para alterar el orden de precedencia

Operaciones del tipo de dato numérico

- Operación de asignación (:=)

- Operaciones de comparación

Además de los operadores matemáticos mencionados, el tipo de dato numérico posee operadores relacionales que permiten comparar valores.

- ✓ **igualdad** (=),
- ✓ **desigualdad** (<>)
- ✓ **orden** (<, <=, >, >=)

Su **resultado** es Verdadero o Falso

- 0 < 7 (verdadero)
- 5 <= 7 (verdadero)
- 5 > 7 (falso)
- 5 <> 7 (verdadero)
- 5 >= 2 (verdadero)
- Pensemos: 5.0 = 5 ???

¿Como se identifican los datos del programa?

IDENTIFICADORES

Para identificar los datos de un programa (constante / variable) se utilizan nombres descriptivos. Esa identificación permite conocer su dirección real en la memoria y el valor que contiene.

En Pascal, los identificadores están formados por letras, dígitos en cualquier orden y algunos símbolos especiales (excepto el primer carácter que **debe ser una letra**).

flores1

papeles

Total_cuadras

pasos

cantidad

Es obligatorio declarar cada uno de los datos que utiliza el programa

Declaración de constantes y variables en Pascal

En **Pascal**, las constantes deben ser declaradas antes de ser usadas.

Siguiendo su notación, la declaración de constantes **en Pascal** se realiza mediante la palabra clave **const**, de la forma:

```
const  nombre = valor
```

donde nombre es el identificador que representa el nombre de la constante. El tipo de dato de la constante queda definido implícitamente por el valor que se le asigna.

```
const
   N = 25          {N se asume entero }
   pi = 3.1416     {pi se asume  real }
```

Declaración de variables

Las variables en Pascal se declaran utilizando la palabra clave **var**, de la forma:

```
var  variable : tipo
```

Si hay varias variables del mismo tipo, se pueden declarar separadas por coma y se le asocia el tipo de dato, una sola vez.

Por ejemplo:

```
var
 variable1: tipo A;
 variable2, variable3: tipoB;
```

Declaración de variables numéricas en Pascal

En Pascal, para representar los datos numéricos se cuenta con:

- **integer** ➡ dato numérico entero
- **real** ➡ dato numérico real

```
Var
   cantidad:  integer;     {cantidad puede contener un nro. entero}
   total_cobrado: real;   {total_cobrado puede contener número real }
```

Cuando se asigna el tipo de dato a una variable queda determinado:

Valores posibles / Operaciones permitidas / Representación en memoria

Estructura general de un Programa en Pascal

```
Program nombre;
Const
   ……
Var
   …..

begin      {Cuerpo del programa}
  ……
  …..      {Instrucciones ejecutables}
End.
```

Zona de Declaración de constantes

Zona de Declaración de variables

Zona de Instrucciones ejecutables

Ejemplo de un Programa en Pascal

```
Program nombre;
Const
  num = 5;
Var
  cantidad:  integer;
  total_cobrado: real;
Begin  { Cuerpo del programa}
  cantidad := 4;
  total_cobrado := 10,50;
End.
```

Zona de Declaración de constantes

Zona de Declaración de variables

Zona de Instrucciones ejecutables

Resolución del ejercicio

Escribir un programa que calcule la superficie de un terreno que tiene 15,5 mts. de frente y 60 mts. de largo e informe el resultado.

Algoritmo:

- *Calcular superficie*
- *Mostrar resultado*

¿Como se calcula la superficie?

¿como se muestra el resultado?

Operación de salida en Pascal: WRITE

- **Write**

se usa para mostrar el contenido de una variable, por defecto en pantalla. Pueden ser de tipo entero, real, char. Los datos a mostrar si son más de uno deben ir separados por coma.

```
Write (a, b);
Write (a);
Write (b);
```

- **También puede utilizarse la instrucción Writeln**

Es similar a la instrucción Write, pero hace que la siguiente sentencia Write muestre el contenido de la variable en la siguiente línea.

Resolución del ejercicio

Escribir un programa que calcule la superficie de un terreno que tiene 15,5 mts. de frente y 60 mts. de largo e informe el resultado.

Algoritmo:

- Calcular superficie
- Mostrar resultado

```
Program superficie;
  var sup: real;
Begin
   sup := 15,5 * 60;
   write ('La superficie es: ', sup);
End.
```

Ejercicio

Escribir un programa que calcule la superficie de un terreno cuyas dimensiones se leen de teclado e informe el resultado.

```
Algoritmo:
- Leer dimensiones
- Calcular superficie
- Mostrar resultado
```

¿como se leen las dimensiones?

Operación de entrada en Pascal: READ

- **Read**

se usa para leer datos (por defecto desde teclado) y asignarlos a las variables correspondientes.

```
Read (a, b, c);
Read (a);
Read (b);
Read (c);
```

- **También puede utilizarse la instrucción Readln**

Es similar a la instrucción Read pero hace que la siguiente sentencia Read comience leyendo en la siguiente línea.

Ejercicio 1

Escribir un programa que calcule la superficie de un terreno cuyas dimensiones (frente y fondo) se leen de teclado e informe el resultado.

Algoritmo:

- Leer dimensiones
- Calcular superficie
- Mostrar resultado

```
Program superficie;
  var frente, fondo, sup: real;
Begin
  readln (frente);
  readln (fondo);
  sup := frente * fondo;
  write ('La superficie es: ', sup);
End.
```

Ejercicio 2

Si se quiere alambrar ese terreno ¿Cuántos metros lineales de alambrado se necesitan?

Modificar el programa anterior, para que también resuelva e informe la cantidad de metros necesarios para alambrarlo

Material de lectura sugerido

- Capítulos 1 y 3 de **Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci**. De Giusti, Armando et al. 1er edición. Prentice Hall 2001.