**Estructura de un programa en Pascal**

1. Todo programa en lenguaje Pascal comienza con la palabra **Program** y el nombre del programa (máx. 8 caracteres, sin espacios en blanco) seguido de un punto y coma.

Ejemplo:  
**Program nomina;**

2. Es conveniente utilizar la cláusula uses **crt** para utilizar las propiedades de la pantalla.

Ejemplo:  
**Uses crt;**

3. A continuación definimos las constantes: son aquellas que tienen un valor que no cambiará en todo el programa. Se comienza con la palabra **Const**.

Ejemplo:  
**Const**  
linea = 66;  
max = 100;  
min = 0;  
n = 30;

4. A continuación definimos nuestros tipos de datos. Hay dos tipos de datos en pascal: los tipos de datos **estándares**, es decir que el compilador Pascal los reconoce automáticamente, como son: **integer, real, boolean, char, byte**. Estos tipos de datos no son necesarios definirlos aquí, sino directamente se asignan a las variables. Los tipos de datos que hay que colocar aquí son aquellos **definidos por el usuario**. Ellos son: los tipos **string** (cadenas de caracteres), **los arreglos** (vectores y matrices), **los registros, los archivos y los apuntadores. Se comienza con la palabra *type*.**

Ejemplo:

**Type**  
**nom : string[40];** —> nom es un tipo de datos string de 40 caracteres.  
  
**alum : array[1..5] of integer;** —> alum es un vector de cinco elementos tipo entero  
  
**nota1 : array[1..30,1..4] of real;**—> nota1 es una matriz de 30 filas y cuatro columnas de tipo real.

**emp : record** —> emp es un registro formado por tres campos: ci, nya y sueldob  
 **ci: string[9];  
 nya : string[40];  
 sueldob : real;  
end;**

**emplea : file of emp;** —> emplea es un archivo conformado por registros tipo emp

**vempleados: array [1..15] of emp;** ---> vempleados es un vector compuesto formado por registros tipo emp

5. Posteriormente definimos las variables, asociadas a los tipos de datos que hemos definido en **Type**, o aquellas que son de tipo **estandar**: enteras (integer), reales (real), booleanas. etc.  
Se comienza con la palabra **Var**.   
Ejemplo:

**Var**  
i,j,k : integer;  
promedio : real;  
nombre : nom; —> Observar la relación de la variable con el tipo de datos  
alumno : alum; definido en el type  
notas : nota1;  
empleado : emplea;  
regemp : emp;

Recordar que en el programa trabajaremos con las variables. Ejemplo:  
**read(nombre);**

6. Si tenemos subprogramas, procedimientos o funciones se definen en este lugar. Comienzan con la palabra **Procedure** y el nombre del procedimiento y termina con la palabra **end** y punto y coma.

Ejemplo:  
**Procedure listar;  
Const  
……  
Type  
……  
Var**……  
**begin**……  
**end;**

Un procedimiento tiene la misma estructura del programa principal. Las variables que se utilizan en el procedimiento sólo sirven para ser utilizadas en el procedimiento (variables locales), mientras que las que hemos definido en el programa principal son globales, es decir pueden utilizarse en cualquier parte del programa.

7. Por último, escribimos el programa principal, el cual comienza con la palabra **begin** y termina con la palabra **end** seguida de un punto. Dentro del **begin** y el **end** final se encuentran todas las instrucciones y llamadas a los procedimientos que hemos definido anteriormente.

Ejemplos:

**Program nombre; {cabecera}**

**Uses**

**Crt; {declaraciones}**

**Const**

**……**

**Type**

**……**

**Var**

**……**

**Procedure listar; {declaración de procedimientos}**

**Begin**

**…….**

**end;**

**begin {Programa principal}**

**writeln( 'Escriba el nombre del estudiante: ');{Visualiza información en pantalla}  
readln(nombre); {Leer un dato del teclado}**

**……..  
listar;{llamada al procedimiento listar}  
……..**

**end. {Final del programa}**

**Program MiPrimerPrograma; {cabecera}**

**Uses**

**Crt; {declaraciones}**

**Const**

**iva =0.10;**

**Type**

**cadena =string[35];**

**Var**

**sueldo :real;**

**numero :integer;**

**nombre :cadena;**

**begin {Programa principal}**

**ClrScr; {Limpia la pantalla}**

**Write ('Escribe tu nombre : '); {Visualiza información en pantalla}**

**ReadLn(nombre);{Leer un dato del teclado}**

**WriteLn ('Bienvenido ', nombre); {Visualiza información en pantalla}**

**Readkey; {Espera la pulsación de una tecla}**

**ClrScr**

**end.**