

República Bolivariana de Venezuela
Ministerio de Educación Superior
Instituto Universitario de Tecnología Valencia
Departamento de Informática



GUIA TEORICA N° 4

LENGUAJE DE PROGRAMACION I

APUNTES: CADENAS DE CARACTERES.

«Nada hagáis por contienda o por vanagloria;
Antes bien con humildad, estimando cada uno
A los demás como superiores a él mismo;
No mirando cada uno por lo suyo propio;
Sino cada cual también por lo de los otros.»
Filipenses 2:3-4

Profesor : DELY M. GIL A.

VALENCIA, OCTUBRE 2007

CADENAS DE CARACTERES

La Una cadena de caracteres (string) es una serie de caracteres cuya longitud (número de caracteres que contiene) puede variar entre 1 a 255 caracteres.
Sintaxis:

<identificador> = string [*limite_superior*];

Las variables de cadena se declaran en la sección Var o Type.

Declaración en Var:

Var

nombre: string[30];

Declaración en Type:

Type

cad30= string[30];

Var

nombre : cad30;

LONGITUD DE UNA CADENA

Una cadena en Turbo Pascal tiene dos longitudes :

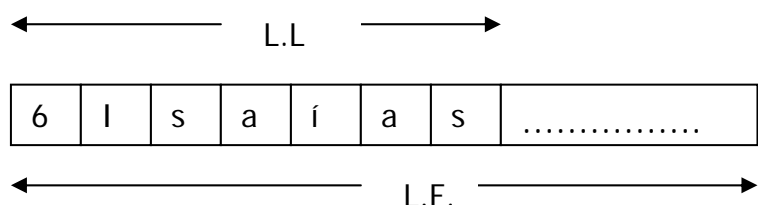
1.Longitud física : Es la cantidad de memoria que ocupa realmente, ésta se establece en tiempo de compilación y nunca cambia. Se define en la sección Type o Var

2.Longitud lógica : Es el número de caracteres almacenados actualmente en la variable cadena. Este dato puede cambiar durante la ejecución del programa.

Ejemplo: Var nombre : string[11];
begin
nombre := 'Susana'
end.

REPRESENTACIÓN DE UNA CADENA DE CARACTERES

La ocupación en memoria de una cadena es un número de bytes igual a la cantidad de caracteres de la cadena más uno, esto se debe a que en el primer byte se almacena la longitud de la cadena.



OPERACIONES ENTRE CADENAS

⊕ Asignación:

Ejemplo: nombre := 'Susana';

⊕ Comparación: Las *comparaciones* de las cadenas de caracteres se hacen según el orden de los caracteres en el código ASCII y con los operadores de relación.

Ejemplo : 'Alex' > 'Alas' {puesto que 'e' > 'a'}

'ADAN' < 'adan' {puesto que 'A' < 'a'}

⊕ Concatenación: Se usa el signo más(+).

'CONTAB' + '.' + 'PAS' = 'CONTAB.PAS'

CADENA FÍSICAMENTE

Una cadena físicamente es una estructura tipo array de caracteres, por lo tanto la forma de acceder es mediante un índice entre corchetes que especifica la posición del carácter dentro de la cadena, así para el ejemplo anterior se tiene:

```
nombre[1] ==> 'l'  
nombre[2] ==> 's'  
nombre[3] ==> 'a'  
nombre[4] ==> 'í'  
nombre[5] ==> 'a'  
nombre[6] ==> 's'
```

FUNCIÓN LENGTH

Proporciona la longitud lógica de una cadena de caracteres y devuelve un valor entero entre 0 y 255. Formato

Length (s) s expresión tipo cadena

Ejemplo :

```
var cad : string[20];
```

```
begin
```

```
cad:='Hola';
```

```
WriteLn(Length ('Hola como estás?'));
```

{devuelve 15}

```
WriteLn(Length (""));
```

{devuelve cero}

```
WriteLn(Length (cad));
```

{devuelve 4}

```
WriteLn(Ord(cad[0]))
```

{devuelve el valor 4}

```
end.
```

FUNCIÓN CONCAT

Permite unir varias cadenas de caracteres devolviendo como resultado una cadena, funciona de igual manera que el operador "+".

Formato

Concat(s1,s2,...,sn) S1,S2...cadenas o variables de caracteres (expresión tipo cadena)

Ejemplo :

Var

```
cad1,cad2 : string[20];
```

```
destino : string[50];
```

```
begin  
  destino:=Concat(cad1,cad2,' : Pascal ')  
end.
```

FUNCIÓN COPY

Devuelve una cadena de caracteres (subcadena) extraída de una cadena. Formato

Copy(s,posición,número)

s cadena (fuente)

posición primer carácter a extraer (tipo entero)

número total de caracteres a extraer (tipo entero)

Nota:

Si posición es mayor que la longitud de S, se devuelve una cadena vacía; si número especifica más caracteres que los indicados desde posición, sólo se devuelve el resto de la cadena.

EJEMPLO COPY

```
ch := 'Esto es una cadena muy larga';  
ch2 := Copy(ch,1,18);  
Write(ch2);      {ch2: 'Esto es una cadena' }  
Ch3 := Copy(Ch,20,12)  {ch3: 'muy larga' }  
Ch4 := Copy(Ch,30,2)   {ch4: ''}
```

FUNCIÓN UPCASE

Permite cambiar letras a mayúsculas. La función Uppcase toma un argumento de tipo CHAR. Formato

Uppcase (ch) Ch Carácter

Ejemplo:

```
Resp := 's';
```

```
Resp:= Uppcase(Resp);  {Resp: 'S'}
```

NOTA: Esta función **NO** convierte a **MAYÚSCULA** una **CADENA DE CARACTERES (STRING) SINO** un **CARÁCTER**.

FUNCIÓN POS

Permite determinar si una cadena está contenida en otra. En este caso, la función devuelve la posición donde comienza la cadena buscada en la cadena fuente, si la cadena no existe, devuelve el resultado 0.

Formato :



Pos (cadena buscada, cadena fuente)

Ejemplo: cad := 'uno dos tres cuatro cinco seis';
WriteLn(Pos('dos', cad));
 {Resulta 5 que es la posición de 'dos'}
WriteLn(Pos('ocho', cad));
 {Resulta 0 no existe la cadena 'ocho'}

EJEMPLO DELETE

```
Cad := 'computer';  
Delete(cad, 4, 6) ;           {resulta com}  
Cad := 'computer';  
Delete(cad, 4, 3) ;           {resulta comer}  
Delete(cad, 6, 3) ;           {resulta comer}  
Delete(cad, 3, 4) ;           {resulta co}
```

PROCEDIMIENTO DELETE

Borra o elimina una subcadena de una cadena contenida en otra cadena de mayor longitud.

Formato **Delete(s, posición, numero)**

s Cadena original o fuente
posición Expresión entera que indica la posición del primer carácter a suprimir.
numero Cantidad de caracteres a suprimir.

Nota: Si posición es mayor que la longitud de la cadena, no se borra ningún carácter.

Si número especifica más caracteres que los existentes desde la posición inicial hasta el final de la cadena, sólo se borran tantos caracteres como estén en la cadena.

PROCEDIMIENTO INSERT

Inserta una subcadena en una cadena.

Formato



Insert (cad1, s, posición)

cad1 cadena a insertar
s cadena donde se inserta
posición carácter a partir del cual se inserta

Ejemplo:

```
cad := 'José Rodríguez';  
Insert ('García ', cad, 6)  
          {resulta 'José Gracia Rodríguez'} .  
Insert('fin', cad, length(Cad));  
          {resulta 'José García Rodríguezfinz'}
```

PROCEDIMIENTO STR

Este procedimiento convierte un número en una cadena Formato



Str (valor ,s)

Valor expresión numérica

s cadena

Ejemplo:

```
Numero := 235.55;  
Str(Numero,cadena);  
Write(cadena);
```

{ Resulta la cadena '2.355000000E+02'}

PROCEDIMIENTO VAL

Este procedimiento convierte una cadena en variable numérica.

Formato



Val (S, variable, código)

S cadena

Variable variable de tipo entero o real

código si la conversión ha podido ser efectuada toma el valor cero(0); en caso contrario contiene la primera posición del primer carácter de la cadena (<>0 = [1..length(s)])

```
Program Ejemplo_Val;  
Var Cad : string;  
    Num : real;  
    Codigo : integer;
```

Begin

```
Cad:= '12e0';
```

```
Val (cad,Num,Codigo);
```

```
If Codigo <> 0 Then
```

```
    WriteLn('Error en la posición :'), cd)
```

```
Else
```

```
    WriteLn(' El número es :'; num:0:2)    {Produce    }
```

```
cad:='22.25';
```

```
Val(cad,num,codigo);
```

```
if codigo=0 then
```

```
    WriteLn(num:0:2)    {Produce    }
```

```
Else
```

```
    WriteLn(codigo);
```

```
cad:='12x45';
```

```
Val(cad,num,codigo);
```

```
if codigo=0 then
```

```
    WriteLn(num1)
```

```
else
```

```
    WriteLn(codigo)    {Produce    }
```

```
End.
```

```
{Guia4_5}Function Inversa(s:string): String;
Var
  Reves : string;
  i      : byte;
Begin
  Reves := "";
  For l:=length(s) downto 1 do
    Reves := Concat(reves,s[l]);
  Inversa := Reves
End;

Function Es_Palindromo(s:string):Boolean;
Begin
  Es_Palindrome := s = Inversa(s)
end;

Function Es_Palindromo2(s:string):Boolean;
Var i,j,cuenta,k :byte;
Begin
  l:= LENGTH(s);
  cuenta := length(s) div 2
  For J:= 1 to cuenta do
    Begin
      If s[i] <> s[j] then
        begin
          Es_Palindrome2 := False;
          exit
        End;
      dec(i);
    End;
  Es_Palindrome := True
end;

Function EsPalindromo3(s:string):Boolean;
Var
  i,j :byte;
Begin
  i:=1; j:= length(s);
  while (i<j) and (s[i] = s[j]) do
    Begin
      Inc(i); dec(j);
    End;
  EsPalindrome2 := s[i]=s[j]
End;
```

```
Function Buscar_Izquierdo(s:string;Ch:char):Byte;
Var
  i:byte;
Begin
  i:= length(s);
  While (i>=1) and (Ch <> s[i]) do
    Dec(i);
  Buscar_Izquierdo := i
End;

Function Buscar_Izquierdo2(s:string;Ch:char):byte;
Var
  i : byte;
Begin
  For i:= length(s) downto 1 do
    If Ch = S[i] Then
      begin
        Buscar_Izquierdo2 := i;
        exit
      End;
  Buscar_Izquierdo2 := 0;
end;

Function Buscar_Izquierdo3(s:string;Ch:char):byte;
Var
  i : byte;
  Band :boolean;
Begin
  l := length(s);
  Band := False;
  Repeat
    If Ch = S[i] Then Band:= True
    Else Dec(i)
  Until (Band) or (i=0);
  Buscar_Izquierdo3:= i
End;
```

II Quiz (Cadenas de Caracteres)

1. Indicar cuál es la salida de cada sentencia donde encuentre funciones y/o procedimientos de cadenas.

<pre> Cad := 'Visual Basic'; Cad1 := Cad; Delete(Cad1, 4, length(Cad) -3); Cad2 := cad1; Cad3 := copy (Cad2, 1 ,2); Cad4 := concat (Cad,Cad3); Cad5 := 'Visual Interdev 6.0'; Punto := '.'; PosDigito := Pos(punto,Cad5) PosDigit := Pos ('Visula, Cad5) PosDigit2 := Pos ('Visual',Cad); If PosDigit = PosDigit2 Then Cadfinal:= Concat(Cad, '.',Cad5) Else If PosDigito = 0 Then Write ('nada') Else Write('encontro en cad5'); Cad1 := 'Java Script'; Cad2 := Uppcase(Cad1); J := Pos('a',cad2); Delete (cad2, j+1, len(PosPunto)-j); Blanco := ' '; Cad3 := concat('Los siguientes: ', cad, blanco, cad5, blanco, cad1); Insert(Cad3,'lenguajes',14); Cad10 := 'se estudiarán para las Certificaciones de Microsoft'; Insert(Cad3,Cad10,23) LenPalabra = length ('nueve'); Str(LenPalabra, Alumnos); Cadfinal:= 'Solo ' + Alumnos + 'estudiaran?'; CadX := Pos ('6.0',Cad5); CadY := 'o.aspx' Cad8:= Concat(CadX,CadY); Cad9 := Insert(Cad8,blanco,4); Cad_num := Copy (Cad9, 1,4); Val(cad9,numero_versión, codigo); If codigo=0 then Write('La versión de VI es:',numero_versión) Else Write('No hay versión sino actualizacion de asp'); </pre>	<pre> ☺ Cad1← Cad1← Cad2← Cad3← Cad4← Cad5← Punto← PosDigito← PosDigit← PosDigit2← Cadfinal← Salida← Cad1← Cad2← J← Cad2← Blanco← Cad3← Cad3← LenPalabra← Alumnos← Cadfinal← CadX← CadY← Cad8← Cad9← Cad_num← Numero_versión← Salida← </pre>
--	--

2. Indicar cuál es la salida después que se ejecute las siguientes sentencias:

```

1.- Cad := 'fantastico';
   Cad1 := Cad;
   delete(Cad1,4,length(Cad)-3);
   Cad2 := Cad1;
   Cad3 := Copy(Cad2,1,2);
2.- Cad1 := 'Julio';
   Cad2 := 'alejandro';
   J := Pos('x',Cad2);
   Aux := Cad2;
   Delete(Cad2, j+1, Length(Cad2) -j);
   Cad3 := Cad2;
   l := Length(Cad3);
   Cad2 := Aux;
   Aux := Cad3;
   Cad3 := Concat(Cad3,Cad1);
   writeln(Cad1, ' ', Cad2, ' ', Cad3, ' ', l, ' ', j);
3.- Cad := 'Picapiedra';
   M := Length(Cad);
   N := Pos(' ',Cad);
   Cad1 := Cad;
   k := Pos(' ', Cad1);
   writeln(M,N,Cad1 :M, Cad2 :N);
4.- Cad := 'ejemplo uno';
   Nuevo := '';
   k := length(Cad);
   for i:=0 to k do
     begin
       Aux := Cad;
       Delete(Cad,2,length(Cad) -1);
       Uno := Cad;
       writeln(uno);
       Cad := Aux;
       Nuevo := Concat(Nuevo, Uno);
       Delete(Cad,1,1);
       writeln(Nuevo, ' ', Cad);
     end;
5.- Cad := 'turbo';
   Cadena[1] := 'm'; Cadena[2] := 'n'; Cadena[3] := 'p';
   Cadena[4] := 'a'; Cadena[5] := 'b'; Cadena[6] := 'c';
   for i:= 1 to 6 do
     writeln('Números de elemento ',i,'es',cadena[i]);
   Cad1 := 'pascal';
   Cadena := Cad1;
   for i:= 1 to 6 do
     writeln(Cad);
   Cad2 := Cad1 + Cadena;
   writeln('la nueva cadena es: ', Cad2);
   Cadena := 'Madrid';
   writeln('la cadena es ahora',Cadena);
6.- Cad:= 'Pepa';
   Cadena := Cad;
   for i:= 1 to 4 do
     Writeln(Cadena[1]);
   writeln;
   Cadena[1]:= 'b'; Cadena[4]:= 'j';
   for i:= 1 to 4 do
     write(Cadena[i]);
   cadena[1]:= 'd';
   Grande:= Cadena + 'musa ';

```