



TEORÍA

Operaciones con cadenas de caracteres

SAP ABAP Programación





Contents

1. Conceptos	3
2. ENCADENANDO CADENAS DE CARACTERES	3
3. SEPARANDO CADENAS DE CARACTERES.....	3
4. BUSCANDO CARACTERES	4
5. REEMPLAZANDO CARACTERES.....	5
6. CALCULANDO LA LONGITUD DE UNA CADENA DE CARACTERES	5
7. TRANSFORMANDO UNA CADENA DE CARACTERES ENMAYÚSCULAS/MINÚSCULAS	6
8. DESPLAZANDO UNA CADENA DE CARACTERES	6
9. COMPARANDO CADENAS DE CARACTERES	6
10. OPERANDO SÓLO CON UNA PARTE DE UNA CADENA	7



1. Conceptos

Se entiende por cadenas de caracteres cualquier objeto de los tipos: **c**, **d**, **n**, **t** y **string**. Si los componentes de una estructura son todos del tipo carácter podemos operar con la estructura como si fuera una única cadena de caracteres.

2. ENCADENANDO CADENAS DE CARACTERES

Usamos la instrucción **CONCATENATE**, por ejemplo:

CONCATENATE s1 s2 **INTO** s3.

Si queremos que las dos cadenas estén separadas por un espacio en blanco ponemos:

CONCATENATE s1 s2 **INTO** s3 **SEPARATED BY SPACE**.

Es conveniente declarar s3 del tipo **string** para no correr el riesgo de quedarse sin espacio al encadenar varios objetos.

```
DATA: s1(20) TYPE c VALUE 'hola',  
      s2(20) TYPE c VALUE 'amigos',  
      s3      TYPE string.
```

3. SEPARANDO CADENAS DE CARACTERES

Para separar cadenas usamos la instrucción **SPLIT**

SPLIT s3 **AT SPACE INTO** s1 s2.



Podemos separar una cadena de caracteres en una tabla. En este caso cada palabra aparecerá en una línea distinta de la tabla:

```
SPLIT s3 AT SPACE INTO TABLE t.
```

En este caso declaramos t como una tabla del tipo string:

```
DATA t TYPE TABLE OF string.
```

4. BUSCANDO CARACTERES

Para buscar un patrón de caracteres (por ejemplo s1) en una cadena de caracteres (por ejemplo s2) usamos la instrucción **SEARCH**

```
DATA: s1 TYPE string VALUE 'amigos',  
      s2 TYPE string VALUE 'hola amigos'.
```

```
SEARCH s2 FOR s1.
```

Si el sistema encuentra el patrón s1 en s2 la variable de sistema sy-subrc guarda un **5** (recuerde que el sistema empieza a contar desde cero). La variable de sistema sy-fdpos guarda la posición en que comienza el patrón s1 dentro de s2.

Podemos buscar una palabra en una cadena de caracteres, por ejemplo:

```
SEARCH s2 FOR 'amigos'.
```

Podemos buscar una cadena que comience por un patrón, por ejemplo:



```
SEARCH s2 FOR 'ami*'
```

Podemos buscar una cadena que finalice por un patrón, por ejemplo:

```
SEARCH s2 FOR '*gos'
```

5. REEMPLAZANDO CARACTERES

Para reemplazar una cadena de caracteres usamos la instrucción REPLACE. El sistema busca el patrón s1 en s3 y si lo encuentra lo sustituye por s2.

```
DATA: s1 TYPE string VALUE 'hola',  
      s2 TYPE string VALUE 'adiós',  
      s3 TYPE string VALUE 'hola amigos'.
```

```
REPLACE s1 WITH s2 INTO s3.
```

6. CALCULANDO LA LONGITUD DE UNA CADENA DE CARACTERES

Para calcular la longitud de una cadena usamos la función **strlen()**, en este caso nos daría un valor de 4, aunque la variable s puede ocupar un máximo de 70 caracteres.

```
DATA: s(70) TYPE c VALUE 'hola',  
      x TYPE i.
```

```
x = strlen(x).
```



7. TRANSFORMANDO UNA CADENA DE CARACTERES ENMAYÚSCULAS/MINÚSCULAS

Usamos la instrucción **TRANSLATE**, por ejemplo:

TRANSLATE s **TO UPPER CASE**.

TRANSLATE s **TO LOWER CASE**.

El primer caso transforma en mayúscula y el segundo en minúscula.

8. DESPLAZANDO UNA CADENA DE CARACTERES

Usamos la instrucción **SHIFT**, por ejemplo:

SHIFT s **LEFT BY 1 PLACES**.

SHIFT s **RIGHT BY 1 PLACES**.

En el primer ejemplo desplazamos s un lugar hacia la izquierda (el valor por defecto) y en el segundo hacia la derecha.

9. COMPARANDO CADENAS DE CARACTERES

Para ver si todos los caracteres de s1 están en s2:



`s1 CO s2`

Para ver si NO todos los caracteres de s1 están en s2:

`s1 CN s2`

Para ver si al menos un carácter de s2 está en s1:

`s1 CA s2`

Lo contrario sería `s1 NA s2`

Para ver si s1 contiene s2

`s1 CS s2`

Lo contrario sería `s1 NS s2`

Para ver si s2 está en s1 (si es cierto la variable sy-fdpos guarda la posición de s2 dentro de s1)

`s1 CP s2`

Lo contrario sería `s1 NP s2`

10. OPERANDO SÓLO CON UNA PARTE DE UNA CADENA

Podemos operar con tan sólo una parte de una cadena de caracteres especificando la posición dentro de la cadena (offset) y el número de caracteres con lo que vamos a operar, por ejemplo:

```
DATA c(4) VALUE 'hola'.
```

```
WRITE c+2(1).
```

En este caso el sistema mostrará la letra l.



Si todos los componentes de una estructura son de tipo carácter también podemos operar con los componentes de la siguiente manera:

```
DATA: BEGIN OF nombre,  
        nombre(20) TYPE c VALUE 'Juan',  
        apellido(20) TYPE c VALUE 'Pérez',  
END OF nombre.
```

```
FIELD-SYMBOLS <apellido> TYPE c.
```

```
ASSIGN nombre+20(20) TO <apellido>.
```

En este caso tan sólo el apellido se asigna al field symbol.

También podemos asignar el apellido de la siguiente forma (ya que

apellido es el segundo componente de la estructura nombre)

```
DATA x TYPE i VALUE 2.
```

```
ASSIGN COMPONENT x OF STRUCTURE nombre TO <apellido>.
```