

INITIALIZE, se utiliza para inicializar variables según su descripción, es decir pondrá a ceros todas las variables numéricas o de edición y a espacios en blanco las alfabéticas y alfanuméricas. No funciona con campos definidos como FILLER, (evidente). Y puede ser muy útil para inicializar tablas completamente cuando nos referimos al nivel mas alto de la misma.

```
...  
WORKING-STORAGE SECTION.  
77  TEXTO PIC X(10) VALUE "HOLA MUNDO".  
77  NUMERO PIC 9(8) VALUE "12345678".  
...  
PROCEDURE DIVISION.  
INICIO.  
    INITIALIZE TEXTO NUMERO.  
...
```

Nota: Después de hacer INITIALIZE el valor de TEXTO será igual a espacios y el de NUMERO igual a ceros.

Es el mismo resultado que utilizar MOVE haciendo uso de las CONSTANTES FIGURATIVAS que vimos en la Introducción de los manuales:

```
MOVE SPACES TO TEXTO.  
MOVE ZEROS TO NUMERO.
```

El resultado es el mismo en ambos casos, pero en el segundo teníamos que saber de que tipo eran las variables para moverles SPACES o ZEROS, mientras que en el primer caso, es el compilador quien se encarga de saber el tipo de la variable.

INSPECT, esta sentencia se utiliza para contar, reemplazar o contar y reemplazar caracteres o grupos de caracteres dentro de un campo. Se puede contar las veces que aparece un caracter, o cambiar todos esos caracteres por otros, etc ...

Esta instrucción tiene formatos diferentes según lo que se desee hacer, así que vamos a ver cada uno de ellos.

Formato 1:

INSPECT *campo1*
TALLYING *variable1* **FOR (CHARACTERS)**
((BEFORE/AFTER) INITIAL) *Cadena1*
(ALL)(LEADING) *Cadena2* ...
(Se puede repetir de nuevo)

Este formato es el utilizado para contar el número de veces que aparece Identificador1 en el campo1 y guardará el valor en la variable1 que previamente hayamos definido en la WORKING.

- CHARACTERS, indica que cuente todos los caracteres del campo incluso los espacios en blanco.
- ALL, indica que tiene que buscar en todos los caracteres del campo, la cadena especificada en Cadena2.
- LEADING, indica que tiene que buscar la cadena especificada en Cadena2, pero solo hasta que encontremos uno diferente, si nada mas empezar es diferente el resultado sera directamente 0.
- BEFORE INITIAL, busca solo hasta que aparezca la cadena especificada como Cadena1.
- AFTER INITIAL, empieza a buscar justo después de la cadena especificada en Cadena1.

Veamos unos ejemplos para salir de dudas. Primero vamos a definir una WORKING-STORAGE.

```
...
WORKING-STORAGE SECTION.
77 TEXTO PIC X(15) VALUE "PAGINA DE COBOL".
77 CONTA PIC 9(8).
...
PROCEDURE DIVISION.
INICIO.
    INSPECT TEXTO TALLYING CONTA FOR CHARACTERS.
...
El valor de conta será de 15 que son los caracteres que tiene la
variable TEXTO.

    INSPECT TEXTO TALLYING CONTA FOR ALL "A".
...
El valor de conta será de 2 que son las veces que aparece la letra A
en la variable TEXTO.

    INSPECT TEXTO TALLYING CONTA FOR LEADING "A".
...
El valor de conta será de 0 porque no aparece ninguna A en el
primer carácter de la variable TEXTO.

    INSPECT TEXTO TALLYING CONTA FOR ALL "A"
        BEFORE INITIAL "N".
...
El valor de conta será de 1 que son las veces que aparece la letra A
en la variable TEXTO hasta la aparición del caracter N.

    INSPECT TEXTO TALLYING CONTA FOR ALL "A"
```

AFTER INITIAL "G".

...

El valor de conta será de 1 que son las veces que aparece la letra A en la variable TEXTO, empezando a contar desde el caracter G.