

FUNCIONES DE GESTIÓN DE CADENAS	OBSERVACIONES
<p align="center"><b>(strcase cadena [cuál])</b></p> <p><i>Devuelve todos los caracteres alfabéticos de una cadena en mayúsculas o minúsculas</i></p>	<ul style="list-style-type: none"> <li>• Si cuál se omite o da como resultado nil, todos los caracteres alfabéticos de cadena se convierten en mayúscula.</li> <li>• Si se suministra cuál y no es nil, todos los caracteres alfabéticos de cadena se convierten en minúsculas.</li> </ul>
<p align="center"><b>(strcat cadena1 [cadena2] ...)</b></p> <p><i>Devuelve una cadena que es la concatenación de varias cadenas</i></p>	<ul style="list-style-type: none"> <li>• Si se especifican varios argumentos cadena, la función devuelve la suma de las longitudes de todos los argumentos.</li> <li>• Si éstos se omiten o se escribe una cadena vacía, el resultado es cero.</li> </ul>
<p align="center"><b>(strlen [cadena] ...)</b></p> <p><i>Devuelve un número entero que indica la cantidad de caracteres de una cadena</i></p>	
<p align="center"><b>(substr cadena inicio [longitud])</b></p> <p><i>Devuelve una subcadena de una cadena</i></p>	<ul style="list-style-type: none"> <li>• La función substr empieza en la posición del carácter inicial de la cadena y abarca los caracteres indicados en longitud.</li> <li>• Si este argumento no se especifica, la subcadena continúa hasta el final de la cadena.</li> <li>• Los argumentos inicio y longitud deben ser enteros positivos. El primer carácter de la cadena es el número 1, lo que difiere de otras funciones que procesan elementos de una lista (como nth y sname) para los que el primer elemento es el 0.</li> </ul>
<p align="center"><b>(wcmatch cadena patrón)</b></p> <p><i>Realiza búsquedas con patrones de comodines en una cadena</i></p>	<ul style="list-style-type: none"> <li>• La función wcmatch compara la cadena con el patrón para comprobar si coinciden. Si es así, se devuelve T; en caso contrario, se devuelve, nil.</li> <li>• Tanto la cadena como el patrón pueden ser cadenas entrecomilladas o variables. Algunos de ellos son: <ul style="list-style-type: none"> <li># (almohadilla) Busca cualquier dígito numérico</li> <li>@ (arroba) Busca cualquier carácter alfabético</li> <li>. (punto) Busca cualquier carácter no alfanumérico</li> <li>(asterisco) Busca cualquier secuencia de caracteres, incluida una vacía, y puede utilizarse en cualquier punto del patrón de búsqueda: al principio, en medio o al final.</li> </ul> </li> <li>• Para más detalles, consultar manual de personalización.</li> </ul>

## EJEMPLOS DE FUNCIONES DE GESTIÓN DE CADENAS

```
(setq a "elemento a("
"elemento a("
(setq b ")")
")"
(setq i "2" j "4")
"4"
```

**La función STRCAT permite componer mensajes para órdenes como print, princ, getdist, getreal, etc:**

```
(setq mensaje (strcat a i "," j b))
"elemento a(2,4)"
```

**( Más adelante se estudiarán las funciones de modificación de tipo, que permitirán que variables que tengan valores numéricos puedan ser tratadas como cadenas de literales ).**

```
(setq mensajeg (strcase mensaje))
"ELEMENTO A(2,4)"
```

```
(setq mensajep (strcase mensajeg t))
"elemento a(2,4)"
```

```
(strlen mensajep)
15
```

```
(substr mensaje 4 8)
"mento a("
```

```
(substr mensaje 4 2)
"me"
```

```
(substr mensaje 4 6)
"mento "
```

**Si se pretende avanzar más allá de la longitud de la cadena, se devuelve esta hasta el final:**

```
(substr mensaje 4 20)
"mento a(2,4)"
```

**Si el argumento de inicio de la subcadena refleja una posición más allá de la longitud de ésta, se devuelve la cadena vacía:**

```
(substr mensaje 20 2)
""
```

```
(wcmatch "Nombre" "N*")
```

T