

Bloque	Acción que representa	Parámetros	Valor por defecto
<b>Ciclo de vida de las transacciones y la simulación</b>			
START $n$	Simular en el modelo hasta contabilizar $n$ transacciones terminadas	$n$ = cantidad de transacciones	
GENERATE $[m] [,d] [,i] [,c] [,p]$	Arribo de transacciones independientes entre sí cada cierto tiempo $m \pm d$ o $m \times d$	<ul style="list-style-type: none"> <li>■ <math>m</math> = valor medio entre arribos (constante o SNA)</li> <li>■ Si <math>d \in N</math>, es el desvío máximo permitido. Si <math>d</math> es una FUNCTION, se multiplica <math>m \times d</math> para calcular el próximo nacimiento</li> <li>■ <math>i</math> = instante en que se genera la primera transacción</li> <li>■ <math>c</math> = cantidad máxima de transacciones a generar</li> <li>■ <math>p</math> = prioridad de la transacción a generar</li> </ul>	<ul style="list-style-type: none"> <li>■ <math>m = 0</math></li> <li>■ <math>d = 0</math></li> <li>■ <math>i = A \pm B</math> o <math>A \times B</math></li> <li>■ <math>c = \infty</math></li> <li>■ <math>p = 0</math> (prioridad mínima)</li> </ul>
ADVANCE $m [,d]$	Realización de una tarea que dura cierto tiempo $m \pm d$ o $m \times d$	<ul style="list-style-type: none"> <li>■ <math>m</math> = valor medio de la duración de la tarea (constante o SNA)</li> <li>■ Si <math>d \in N</math>, es el desvío máximo permitido. Si <math>d</math> es una FUNCTION, se multiplica <math>m \times d</math> para calcular la duración. <i>No puede ser referencia a FUNCTION</i></li> <li>■ Debe cumplirse que <math>m \geq d</math></li> </ul>	<ul style="list-style-type: none"> <li>■ <math>m = 0</math></li> <li>■ <math>d = 0</math></li> </ul>
TRANSFER $[p, [a]], b$	Bifurcar estocásticamente. Con probabilidad $p$ se bifurca a $b$ , y con probabilidad $1 - p$ se bifurca a $a$ . Si $a$ no se especifica, bifurca al bloque siguiente con probabilidad $1 - p$ . Si ni $p$ ni $a$ se especifican, bifurca siempre a $b$	<ul style="list-style-type: none"> <li>■ <math>0 \leq p &lt; 1</math>, o <math>p</math> es una referencia a una función de <math>n</math> puntos tal que <math>0 \leq y_i &lt; 1000</math> (<math>i = 1..n</math>)</li> <li>■ <math>a</math> y <math>b</math> son rótulos o referencias a funciones</li> </ul>	$p = 1$
TERMINATE $[n]$	Salida de $n$ transacciones. Si $n$ no se especifica, no se contabiliza la transacción pero la misma se destruye	$n \in N$	$n = 0$

Bloque			Acción que representa	Parámetros	Valor por defecto
<b>Facilities, colas, storages y prioridades</b>					
SEIZE	f		Tomar un recurso de uso exclusivo ( <i>facility</i> )	$f$ = identificador de la <i>facility</i> (número o nombre)	
RELEASE	f		Dejar un recurso de uso exclusivo ( <i>facility</i> )	$f$ = identificador de la <i>facility</i> (número o nombre)	
PRIORITY	p		Cambia la prioridad de la transacción activa	$p$ = nueva prioridad	
NSTORAGE	STORAGE	n	Definir la capacidad de un <i>storage</i> en $n$ transacciones	<ul style="list-style-type: none"> <li>■ <math>NSTORAGE</math> = identificador del <i>storage</i> (número o nombre)</li> <li>■ <math>n \in N, n &lt; 30000</math></li> </ul>	
ENTER	s[,1]		Intentar ocupar uno o más lugares de un recurso de uso compartido ( <i>storage</i> )	<ul style="list-style-type: none"> <li>■ <math>s</math> = identificador del <i>storage</i> (número o nombre)</li> <li>■ <math>l</math> = lugares que ocupa la transacción</li> </ul>	$l = 1$
LEAVE	s[,1]		Dejar uno o más lugares de un recurso de uso compartido ( <i>storage</i> ). No es necesario haber ejecutado ENTER previamente.	<ul style="list-style-type: none"> <li>■ <math>s</math> = identificador del <i>storage</i> (número o nombre)</li> <li>■ <math>l</math> = lugares que deja la transacción</li> </ul>	$l = 1$
QUEUE	f[,1]		Hacer cola para la <i>facility</i> $f$	<ul style="list-style-type: none"> <li>■ <math>f</math> = identificador de la <i>queue</i> (número o nombre)</li> <li>■ <math>l</math> = lugares que ocupa la transacción</li> </ul>	$l = 1$
DEPART	f[,1]		Salir de la cola de la <i>facility</i> $f$ . Si la QTABLE respectiva está definida, registra allí las estadísticas	<ul style="list-style-type: none"> <li>■ <math>f</math> = identificador de la <i>queue</i> (número o nombre)</li> <li>■ <math>l</math> = lugares que deja la transacción</li> </ul>	$l = 1$

Bloque			Acción que representa	Parámetros	Valor por defecto
Tabulación de datos					
NTABLE	TABLE	$v, x_0, \Delta x, n$	Definir una tabla de distribución de frecuencia. Necesita de TABULATE.	<ul style="list-style-type: none"><li>▪ <math>v</math> = valor a tabular de la transacción que ejecuta el <i>tabulate</i> (cualquier SNA)</li><li>▪ <math>x_0</math> = Límite superior del primer intervalo</li><li>▪ <math>\Delta x</math> = tamaño de cada intervalo</li><li>▪ <math>n</math> =cantidad de intervalos</li></ul>	
	TABULATE	t	Agrega un valor a la TABLE t	$t$ = identificador de la TABLE	
NQTABLE	QTABLE	$ntable, t_0, \Delta t, n$	Define una tabla de distribución de frecuencias para tiempos en cola. No necesita de TABULATE.	<ul style="list-style-type: none"><li>▪ <math>NQTABLE</math> = identificador de la <i>queue</i> (número o nombre)</li><li>▪ <math>t_0</math> = límite superior del primer intervalo</li><li>▪ <math>\Delta t</math> = tamaño de cada intervalo</li><li>▪ <math>n</math> = cantidad de intervalos</li></ul>	
Casilleros de memoria y parámetros de las transacciones					
	SAVEVALUE	$s, v$ $s+, v$ $s-, v$	Asigna, suma o resta el valor $v$ al <i>savevalue</i> $s$	<ul style="list-style-type: none"><li>▪ <math>s</math> = identificador del <i>savevalue</i> (número o nombre)</li><li>▪ <math>v</math> = valor a asignar, sumar o restar</li></ul>	
	INITIAL	$s, a$	Asigna el valor inicial $a$ al <i>savevalue</i> $s$	<ul style="list-style-type: none"><li>▪ <math>s</math> = identificador del <i>savevalue</i> (número o nombre)</li><li>▪ <math>v</math> = valor con el que se inicializa</li></ul>	
	ASSIGN	$p, v$ $p+, v$ $p-, v$	Asigna, suma o resta el valor $v$ al parámetro $p$ de la transacción activa	<ul style="list-style-type: none"><li>▪ <math>p</math> = identificador del parámetro (puede ser un SNA)</li><li>▪ <math>v</math> = valor a asignar, sumar o restar (constante o SNA)</li></ul>	
	MARK	[p]	Se almacena un M1 “paralelo” en el parámetro $p$ de la transacción. Si $p$ no se especifica, se reinicializa M1 en cero.	$p$ = id del parámetro	$p = M1$

Bloque	Acción que representa	Parámetros	Valor por defecto
<b>Selección de <i>facilities</i> y objetos</b>			
SELECT      op logico <sup>1</sup> p,v <sub>1</sub> ,v <sub>2</sub> [,,f]	Busca una <i>facility</i> , <i>storage</i> o <i>logic switch</i> según el op logico y la almacena en el parámetro <i>p</i> de la transacción activa. Al finalizar, $v_1 \leq a \leq v_2$ o, si no pudo seleccionar, bifurca a <i>f</i> . Si <i>f</i> no esta, establece que $p = 0$ y continúa en el bloque siguiente	<ul style="list-style-type: none"> <li>■ <i>p</i> = identificador del parámetro</li> <li>■ <i>v<sub>1</sub></i> = número de objeto menor desde el cual se hace el SELECT</li> <li>■ <i>v<sub>2</sub></i> = número de objeto mayor desde el cual se hace el SELECT</li> <li>■ <i>f</i> = rótulo al que bifurca si la selección fue infructuosa</li> </ul>	
SELECT      op relacion <sup>2</sup> a,v <sub>1</sub> ,v <sub>2</sub> ,d,e[,f]	Busca una objeto <i>e</i> según el op relacion y lo almacena en el parámetro <i>a</i> de la transacción activa. Al finalizar, $v_1 \leq a \leq v_2$ o, si no hay desocupadas, bifurca a <i>f</i> . Si <i>f</i> no esta, establece que $a = 0$ y continúa en el bloque siguiente	<ul style="list-style-type: none"> <li>■ <i>a</i> = identificador del parámetro</li> <li>■ <i>v<sub>1</sub></i> = número de objeto menor desde el cual se hace el SELECT</li> <li>■ <i>v<sub>2</sub></i> = número de objeto mayor desde el cual se hace el SELECT</li> <li>■ <i>d</i> = valor con el que se compara</li> <li>■ <i>e</i> = clase de objeto que se compara (SNAs)</li> <li>■ <i>f</i> = rótulo al que bifurca si la selección fue infructuosa</li> </ul>	
SELECT      MIN/MAX      p,v <sub>1</sub> ,v <sub>2</sub> ,,e	Busca un <i>facility</i> desocupado según el criterio <i>e</i> y lo almacena en <i>p</i> . Si el criterio no alcanza para definir, $p = v_1$	<ul style="list-style-type: none"> <li>■ <i>p</i> = identificador del parámetro</li> <li>■ <i>v<sub>1</sub></i> = número menor del objeto a consultar</li> <li>■ <i>v<sub>2</sub></i> = número mayor del objeto a consultar</li> <li>■ <i>e</i> = clase de objeto en la que se busca el mínimo o máximo</li> </ul>	

<sup>1</sup>Logic Switch: LS, LR

Facility: U, NU, I, NI, FV, FNV

Storage: SE, SNE, SF, SNF, SV, SNV

<sup>2</sup>EQ, L, LE, G, GE, NE

Bloque			Acción que representa	Parámetros	Valor por defecto												
Variables, funciones, ciclos y tests																	
NVARIABLE	VARIABLE	f	Define una función que se evalúa cada vez que una transacción hace referencia a ella. Se truncan los decimales de los resultados intermedios y del resultado final.	<div><div>f = función con operadores (cualquier SNA)</div><table><tr><td>+</td><td>suma</td></tr><tr><td>-</td><td>resta</td></tr><tr><td>#</td><td>multiplicación</td></tr><tr><td>/</td><td>división</td></tr><tr><td>@</td><td>módulo</td></tr><tr><td>^</td><td>potencia</td></tr></table></div>	+	suma	-	resta	#	multiplicación	/	división	@	módulo	^	potencia	
+	suma																
-	resta																
#	multiplicación																
/	división																
@	módulo																
^	potencia																
NFUNCTION	FUNCTION	x,tn x <sub>1</sub> ,y <sub>1</sub> /.../x <sub>n</sub> ,y <sub>n</sub>	Devuelve un valor cada vez que se la invoca. Define una función de n puntos. Si x ≤ x <sub>1</sub> , devuelve y <sub>1</sub> , si x ≥ x <sub>n</sub> , devuelve y <sub>n</sub> . Se truncan los decimales de los resultados intermedios y del resultado final.	<ul style="list-style-type: none"><li>■ x = variable independiente (cualquier SNA). Si x es RN, 0 &lt; x<sub>i</sub> ≤ 1</li><li>■ t = tipo de la función (C =continua, D =discreta, E =discreta de atributos numéricos)<sup>3</sup></li><li>■ n = cantidad de puntos que definen la función</li></ul>													
	LOOP	p,a	Resta 1 al parámetro p. Si p > 0, bifurca al rótulo a. Si p = 0, continúa en el bloque siguiente	<ul style="list-style-type: none"><li>■ p = id del parámetro a decrementar</li><li>■ a = rótulo al que bifurca</li></ul>													
TEST	op	a,b[,f]	Si a op b es verdadera, continúa en el bloque siguiente. Si es falsa, bifurca al rótulo f; si es falsa y no se especifico f, bloquea la transacción hasta que sea verdadera	<ul style="list-style-type: none"><li>■ op puede ser e, ne, g, ge, l, le</li><li>■ a y b pueden ser constantes o SNAs</li><li>■ f = rótulo al que bifurca si es falsa</li></ul>													

<sup>3</sup>Debe cumplirse que  $x_1 \leq x_2 \leq \dots \leq x_n$ .

- Si es **continua**, y  $x_i < x < x_{i+1}$  se interpola linealmente entre  $y_i$  e  $y_{i+1}$ .
- Si es **discreta**, y  $x_i < x < x_{i+1}$ , se devuelve  $y_{i+1}$  (los  $y$  pueden ser rótulos).
- Si es **discreta de atributos numéricos**, es como la discreta pero las  $y$  deben ser SNAs.

Bloque		Acción que representa	Parámetros	Valor por defecto	
Familias de transacciones					
SPLIT	$c, r[, n]$	Genera $c$ nuevas transacciones, “clonando ” la transacción activa (M1, PR, parámetros). Las copias bifurcan a $r$ y el original continúa en el bloque siguiente	<ul style="list-style-type: none"><li>▪ <math>c</math> = cantidad de copias (constante o SNA)</li><li>▪ <math>r</math> = rótulo al que bifurcan las copias</li><li>▪ <math>n</math> = identificador de parámetro a utilizar para enumerar el original y las copias</li></ul>		
ASSEMBLE	$a$	Reúne un grupo de transacciones y deja pasar solo una transacción que representa al grupo, las restantes $a - 1$ son destruidas	$a$ = cantidad de miembros de la familia a sincronizar		
GATHER	$m$	Reúne un grupo de transacciones y deja pasar a todas una vez reunida la cantidad especificada, sin destruir ninguna transacción	$m$ = cantidad de miembros de la familia a reunir		
Matrices					
NMATRIX	MATRIX	$, f, c$	Define una matriz	<ul style="list-style-type: none"><li>▪ <math>f</math> = cantidad de filas</li><li>▪ <math>c</math> = cantidad de columnas</li></ul>	
MSAVEVALUE	$m, f, c, v$ $m+, f, c, v$ $m-, f, c, v$	Asigna, suma o resta el valor $v$ del elemento $m[f, c]$	<ul style="list-style-type: none"><li>▪ <math>m</math> = id de la matriz</li><li>▪ <math>f</math> = fila (cualquier SNA)</li><li>▪ <math>c</math> = columna (cualquier SNA)</li><li>▪ <math>v</math> = valor a asignar, sumar o restar</li></ul>		

Bloque			Acción que representa	Parámetros	Valor por defecto
LOGIC	v	ll	Cambia el estado del <i>logic switch</i> ll por el valor de v	<ul style="list-style-type: none"> <li>■ v = S (set), R (reset), I (invertido)</li> <li>■ ll = id de la llave</li> </ul>	
TRANSFER	SIM, b, c		Si el <i>delay indicator</i> de la transacción está en ON (set) —si alguno de los GATE/TEST anteriores no se cumple—, la transacción es enviada al rótulo c y pone el <i>delay indicator</i> en OFF (reset). Si el <i>delay indicator</i> está en OFF —todos los GATE/TEST anteriores se cumplen <b>simultáneamente</b> —, la transacción es enviada al rótulo b.	b y c son rótulos.	
INITIAL	rll[,v]		Inicializa el estado de la llave rll con el valor v.	<ul style="list-style-type: none"> <li>■ rll = <b>referencia</b> a una llave lógica</li> <li>■ v = 1 (set), 0 (reset)</li> </ul>	v = 1(set)
GATE	op lógico <sup>4</sup>	a[,f]	Consulta el estado del <i>logic switch</i> , <i>facility</i> , <i>storage</i> o rótulo a. Si op A es V, continúa en el bloque siguiente. Si es F, bifurca a f. Si no se especifica f, la transacción se bloquea hasta que sea V ( <i>delay indicator</i> en ON)	<ul style="list-style-type: none"> <li>■ a = id del <i>logic switch</i>, <i>facility</i>, <i>storage</i></li> <li>■ f = rótulo al que bifurca si la condición es falsa.</li> </ul>	
COUNT	op lógico <sup>4</sup>	a, v <sub>1</sub> , v <sub>2</sub>	Cuenta la cantidad de objetos desde v <sub>1</sub> hasta v <sub>2</sub> que cumplen la condición <i>op objeto</i> , y lo almacena en a.	<ul style="list-style-type: none"> <li>■ a = parámetro que recibe la cantidad de objetos que cumplen la condición</li> <li>■ v<sub>1</sub> = número menor del objeto a consultar</li> <li>■ v<sub>2</sub> = número mayor del objeto a consultar</li> </ul>	

<sup>4</sup>Logic Switch: LS, LR

Facility: U, NU, I, NI, FV, FNV

Storage: SE, SNE, SF, SNF, SV, SNV

Bloque	Acción que representa		Parámetros	Valor por defecto
COUNT      op relacion <sup>5</sup> a, v <sub>1</sub> , v <sub>2</sub> , d, e	Cuenta la cantidad de objetos desde v <sub>1</sub> hasta v <sub>2</sub> que cumplen la condición <i>objeto op d</i> , y lo almacena en a.		<ul style="list-style-type: none"> <li>■ a = parámetro que recibe la cantidad de objetos que cumple la condición</li> <li>■ v<sub>1</sub> = número menor del objeto a consultar</li> <li>■ v<sub>2</sub> = número mayor del objeto a consultar</li> <li>■ d = valor con el que se compara</li> <li>■ e = clase de objeto que se selecciona (SNA)</li> </ul>	
COUNT      MIN/MAX      a, v <sub>1</sub> , v <sub>2</sub> , , e	Cuenta la cantidad de objetos desde v <sub>1</sub> hasta v <sub>2</sub> que son mínimo o máximo, y lo almacena en a.		<ul style="list-style-type: none"> <li>■ a = parámetro que recibe la cantidad de objetos que son mínimo o máximo</li> <li>■ v<sub>1</sub> = número menor del objeto a consultar</li> <li>■ v<sub>2</sub> = número mayor del objeto a consultar</li> <li>■ e = clase de objeto que se compara (SNA)</li> </ul>	
JOIN      g	La transacción activa se une al grupo g “personalmente”.		g = id del grupo.	
EXAMINE      g, , f	Verifica si la transacción activa pertenece al grupo g. Si pertenece, continúa en el bloque siguiente. Si no, bifurca a f.		<ul style="list-style-type: none"> <li>■ g = id del grupo.</li> <li>■ f = rótulo al que bifurca.</li> </ul>	
REMOVE      g	La transacción activa se elimina al grupo g “personalmente”.		g = id del grupo.	

<sup>5</sup>EQ, L, LE, G, GE, NE



Bloque	Acción que representa	Parámetros	Valor por defecto
<b>REMOVE</b> $[op]^6$ $g, [c], , [p], [v], [f]$	Intenta eliminar $c$ transacciones del grupo $g$ que cumplen la condición $p op v$ . Si no puede eliminar todas, bifurca a $f$ .	<ul style="list-style-type: none"> <li>■ <math>g</math> = id del grupo</li> <li>■ <math>c</math> = cantidad de transacciones a eliminar</li> <li>■ <math>p</math> = id del parámetro que se consulta de la transacción</li> <li>■ <math>v</math> = valor con el que se compara (no se pone si <math>op = MIN/MAX</math>)</li> <li>■ <math>f</math> = rótulo al que bifurca si no pudo eliminar <math>c</math> transacciones</li> </ul>	$op = E$ $c = ALL$
<b>ALTER</b> $[op]^7$ $g, c, p_1, v_1, [p_2], [v_2], [f]$	Intenta alterar $c$ transacciones del grupo $g$ de la siguiente forma: para cada transacción, si $p_2 op v_2$ es V, se asigna $p_1 = v_1$ . Si no puede alterar $c$ transacciones, bifurca al rótulo $f$ .	<ul style="list-style-type: none"> <li>■ <math>g</math> = id del grupo</li> <li>■ <math>c</math> = cantidad de transacciones a eliminar<sup>8</sup></li> <li>■ <math>p_1</math> = id del parámetro a alterar</li> <li>■ <math>v_1</math> = valor a asignar al parámetro <math>p_1</math> (constante o SNA)</li> <li>■ <math>p_2</math> = id del parámetro a consultar</li> <li>■ <math>v_2</math> = valor con el que se compara <math>p_2</math> (constante o SNA)</li> <li>■ <math>f</math> = rótulo al que bifurca si no pudo alterar <math>c</math> transacciones</li> </ul>	$op = E$
<b>SCAN</b> $[op]$ $g, b, [c], p_1, p_2, [f]$	Busca en el grupo $g$ la <b>primera</b> transacción que cumple la condición $b op c$ . Copia el parámetro $p_1$ en el parámetro $p_2$ de la transacción actual. Si no encuentra ninguna, bifurca a $f$ .	<ul style="list-style-type: none"> <li>■ <math>OP</math> puede ser <math>e, g, ge, l, le, max, min, ne</math></li> <li>■ <math>b</math> = id del parámetro que se consulta</li> <li>■ <math>c</math> = valor con el que se compara (si <math>OP</math> es <math>min/max</math>, no se pone)</li> <li>■ <math>p_1</math> = id del parámetro de la transacción del grupo que se copia</li> <li>■ <math>p_2</math> = id del parámetro de la transacción actual que recibe el valor de <math>p_1</math></li> </ul>	<ul style="list-style-type: none"> <li>■ <math>op = E</math></li> </ul>

<sup>6</sup>E, G, GE, L, LE, MAX, MIN, NE<sup>7</sup>E, G, GE, L, LE, NE<sup>8</sup> $c$  puede ser ALL

Bloque			Acción que representa	Parámetros	Valor por defecto
LINK	$c, r[, f]$		Encadena la transacción pasiva actual a la cadena $c$ según la regla $r$ . Si se especifica el rótulo $f$ , la primera transacción <b>no es encadenada</b> y bifurca al rótulo $f$ .	<ul style="list-style-type: none"> <li>■ <math>c</math> = id de la cadena (constante o SNA)</li> <li>■ <math>r</math> = regla por la cual se encadena (FIFO, LIFO, PR, Pn, M1)</li> <li>■ <math>f</math> = rótulo al que va la primera transacción</li> </ul>	
UNLINK	$[op]$	$a, b, c, [d], [e], [f]$	La transacción activa actual desencadena $c$ transacciones de la cadena $a$ tales que se verifique $dope$ , y las envía al rótulo $b$ . Si no puede liberar todas, bifurca a $f$ (las que fueron liberadas, quedan liberadas).	<ul style="list-style-type: none"> <li>■ <math>op</math> puede ser E, G, GE, L, LE, NE</li> <li>■ <math>a</math> = id de la cadena</li> <li>■ <math>b</math> = rótulo al que se envían las transacciones liberadas</li> <li>■ <math>c</math> = cantidad de transacciones que se desencadenan<sup>9</sup></li> <li>■ <math>d</math> = id de parámetro de la transacción <b>activa</b> que se consulta</li> <li>■ <math>e</math> = <b>referencia</b> del parámetro de la transacción <b>pasiva</b> que se consulta</li> <li>■ <math>f</math> = rótulo al que bifurca la transacción activa si no pudo liberar <math>c</math> transacciones.</li> </ul>	<ul style="list-style-type: none"> <li>■ <math>op = E</math></li> </ul>
BUFFER			La transacción actual se coloca como última transacción a mover en la CEC. Detiene a la misma sin que pase el tiempo.		
PREEMPT	$a$		La transacción actual arrebató la <i>facility</i> $a$ que ha sido tomada por un SEIZE, sin importar las prioridades. Si fue tomada por otro PREEMPT, <b>no</b> la arrebató. Interrumpe el ADVANCE del <i>owner</i> de la <i>facility</i> .	$a$ = id de <i>facility</i> a intentar arrebató	
PREEMPT	$a, PR[, c, d[, RE]]$		El arrebato se produce sólo si el <i>owner</i> de la <i>facility</i> $a$ tiene menos prioridad que la transacción actual. Si se especifica RE, el arrebato es definitivo.	$a$ = id de <i>facility</i> a intentar arrebató $c$ = rótulo al que bifurca el <i>owner</i> si estaba ejecutando un ADVANCE $d$ = id del parámetro del <i>owner</i> donde se almacena el tiempo remanente del <i>owner</i> para terminar el ADVANCE	

<sup>9</sup> $c$  puede ser ALL

Bloque		Acción que representa	Parámetros	Valor por defecto
RETURN	a	Retorno de recurso <i>a</i> al <i>owner</i> de dicha <i>facility</i>		
FUNAVAIL	a, [b], [c], [d], [e], f, [g]	Rotura de <i>facility a</i> . El <i>owner</i> de la <i>facility</i> bifurca al rótulo <i>c</i> . Las transacciones que fueron interrumpidas bifurcan al rótulo <i>f</i> . Las transacciones que están en la <i>delay chain</i> o <i>pending chain</i> de <i>a</i> bifurcan al rótulo <i>h</i> .	<ul style="list-style-type: none"> <li>■ <i>a</i> = id de <i>facility</i></li> <li>■ <i>b</i> = <i>RE</i> (el <i>owner</i> de la <i>facility</i> la deja definitivamente) o <i>CO</i> (el <i>owner</i> continúa en poder de la <i>facility</i>)</li> <li>■ <i>c</i> es obligatorio si <i>b</i> = <i>RE</i></li> <li>■ <i>d</i> = parámetro del <i>owner</i> de la <i>facility</i> donde se guarda el tiempo remanente del <i>ADVANCE</i> que estuviera ejecutando.</li> <li>■ <i>e</i> = <i>RE</i> (las transacciones interrumpidas la dejan definitivamente) o <i>CO</i> (el <i>owner</i> continúa en poder de la <i>facility</i>)</li> <li>■ <i>f</i> es obligatorio si <i>e</i> = <i>RE</i></li> <li>■ <i>g</i> = <i>RE</i> (las transacciones en la <i>delay chain</i> o <i>pending chain</i> abandonan la idea de tomar <i>a</i>) o <i>CO</i> (abandonan la <i>delay chain</i> o <i>pending chain</i>)</li> <li>■ si <i>h</i> es usado, <i>g</i> debe ser <i>RE</i></li> </ul>	
FAVAIL	a	Arreglo de <i>facility a</i>		
SUNAVAIL	a	Rotura de <i>storage a</i>		
SAVAIL	a	Arreglo de <i>storage a</i>		
MATCH	a	Sincronizar el movimiento de transacciones de la misma familia.	<i>a</i> = rótulo para ser testeado por transacciones iguales.	
JOIN	g,n	Unir el número <i>n</i> al grupo numérico <i>g</i>	<ul style="list-style-type: none"> <li>■ <i>g</i>= id del grupo <b>numérico</b> o SNA</li> <li>■ <i>n</i> =constante o SNA</li> </ul>	
REMOVE	g,,n[,,,f]	Remover al número <i>n</i> del grupo numérico <i>g</i> . Si no pertenece, bifurca a <i>f</i>	<ul style="list-style-type: none"> <li>■ <i>g</i>= id del grupo <b>numérico</b> o SNA</li> <li>■ <i>n</i> =constante o SNA</li> </ul>	
EXAMINE	g,n,f	Si el número <i>n</i> pertenece al grupo numérico <i>g</i> , continúa en el bloque siguiente. Si no, bifurca a <i>f</i>	<ul style="list-style-type: none"> <li>■ <i>g</i>= id del grupo <b>numérico</b> o SNA</li> <li>■ <i>n</i> =constante o SNA</li> </ul>	

Para referirse a parámetros:

1. Por nombre:

- a)* P\$CAJAS
- b)* \*\$CAJAS
- c)* \*CAJAS

2. Por número:

- a)* P4
- b)* \*4

Para referirse a un SAVEVALUE:

1. Por nombre: X\$CAJAS

2. Por número: X4

Para referirse a una VARIABLE:

1. Direccionamiento directo: V\$CAJA, V\$1

2. Direccionamiento indirecto: V\*CAJA, V\*1

Para referirse a una FUNCTION: FN\$FUNCION

Para referirse a una MATRIX:

1. Por nombre: MX\$MATRIZ(col,fil)

2. Por número: Mx4(col,fil)