

Bloque	Acción que representa	Parámetros	Valor por defecto
Ciclo de vida de las transacciones y la simulación			
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"> ■ m = valor medio entre arribos (constante o SNA) ■ Si $d \in N$, es el desvío máximo permitido. Si d es una FUNCTION, se multiplica $m \times d$ para calcular el próximo nacimiento ■ i = instante en que se genera la primera transacción ■ c = cantidad máxima de transacciones a generar ■ p = prioridad de la transacción a generar 	<ul style="list-style-type: none"> ■ $m = 0$ ■ $d = 0$ ■ $i = A \pm B$ o $A \times B$ ■ $c = \infty$ ■ $p = 0$ (prioridad mínima)
ADVANCE $m [,d]$	Realización de una tarea que dura cierto tiempo $m \pm d$ o $m \times d$	<ul style="list-style-type: none"> ■ m = valor medio de la duración de la tarea (constante o SNA) ■ Si $d \in N$, es el desvío máximo permitido. Si d es una FUNCTION, se multiplica $m \times d$ para calcular la duración. <i>No puede ser referencia a FUNCTION</i> ■ Debe cumplirse que $m \geq d$ 	<ul style="list-style-type: none"> ■ $m = 0$ ■ $d = 0$
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"> ■ $0 \leq p < 1$, o p es una referencia a una función de n puntos tal que $0 \leq y_i < 1000$ ($i = 1..n$) ■ a y b son rótulos o referencias a funciones 	$p = 1$
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
Facilities, colas, storages y prioridades					
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"> ■ $NSTORAGE$ = identificador del <i>storage</i> (número o nombre) ■ $n \in N, n < 30000$ 	
ENTER	s[,l]		Intentar ocupar uno o más lugares de un recurso de uso compartido (<i>storage</i>)	<ul style="list-style-type: none"> ■ s = identificador del <i>storage</i> (número o nombre) ■ l = lugares que ocupa la transacción 	$l = 1$
LEAVE	s[,l]		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"> ■ s = identificador del <i>storage</i> (número o nombre) ■ l = lugares que deja la transacción 	$l = 1$
QUEUE	f[,l]		Hacer cola para la <i>facility</i> f	<ul style="list-style-type: none"> ■ f = identificador de la <i>queue</i> (número o nombre) ■ l = lugares que ocupa la transacción 	$l = 1$
DEPART	f[,l]		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"> ■ f = identificador de la <i>queue</i> (número o nombre) ■ l = lugares que deja la transacción 	$l = 1$

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"> ■ v = valor a tabular de la transacción que ejecuta el <i>tabulate</i> (cualquier SNA) ■ x_0 = Límite superior del primer intervalo ■ Δx = tamaño de cada intervalo ■ n = cantidad de intervalos 	
	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"> ■ $NQTABLE$ = identificador de la <i>queue</i> (número o nombre) ■ t_0 = límite superior del primer intervalo ■ Δt = tamaño de cada intervalo ■ n = cantidad de intervalos 	
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"> ■ s = identificador del <i>savevalue</i> (número o nombre) ■ v = valor a asignar, sumar o restar 	
	INITIAL	s, a	Asigna el valor inicial a al <i>savevalue</i> s	<ul style="list-style-type: none"> ■ s = identificador del <i>savevalue</i> (número o nombre) ■ v = valor con el que se inicializa 	
	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"> ■ p = identificador del parámetro (puede ser un SNA) ■ v = valor a asignar, sumar o restar (constante o SNA) 	
	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
Selección de <i>facilities</i> y objetos			
SELECT op logico ¹ p,v ₁ ,v ₂ [,,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 <i>p</i> = 0 y continúa en el bloque siguiente	<ul style="list-style-type: none"> ■ <i>p</i> = identificador del parámetro ■ <i>v</i>₁ = número de objeto menor desde el cual se hace el SELECT ■ <i>v</i>₂ = número de objeto mayor desde el cual se hace el SELECT ■ <i>f</i> = rótulo al que bifurca si la selección fue infructuosa 	
SELECT op relacion ² a,v ₁ ,v ₂ ,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 <i>a</i> = 0 y continúa en el bloque siguiente	<ul style="list-style-type: none"> ■ <i>a</i> = identificador del parámetro ■ <i>v</i>₁ = número de objeto menor desde el cual se hace el SELECT ■ <i>v</i>₂ = número de objeto mayor desde el cual se hace el SELECT ■ <i>d</i> = valor con el que se compara ■ <i>e</i> = clase de objeto que se compara (SNAs) ■ <i>f</i> = rótulo al que bifurca si la selección fue infructuosa 	
SELECT MIN/MAX p,v ₁ ,v ₂ ,,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, <i>p</i> = <i>v</i> ₁	<ul style="list-style-type: none"> ■ <i>p</i> = identificador del parámetro ■ <i>v</i>₁ = número menor del objeto a consultar ■ <i>v</i>₂ = número mayor del objeto a consultar ■ <i>e</i> = clase de objeto en la que se busca el mínimo o máximo 	

¹Logic Switch: LS, LR

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

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

²EQ, 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><div><div>f = función con operadores (cualquier SNA)</div><div><div><div>+</div><div>suma</div></div><div><div>-</div><div>resta</div></div><div><div>#</div><div>multiplicación</div></div><div><div>/</div><div>división</div></div><div><div>@</div><div>módulo</div></div><div><div>^</div><div>potencia</div></div></div></div></div></div>	
NFUNCTION	FUNCTION	x,tn x1,y1/.../xn,yn	Devuelve un valor cada vez que se la invoca. Define una función de n puntos. Si x ≤ x1, devuelve y1, si x ≥ xn, devuelve yn. Se truncan los decimales de los resultados intermedios y del resultado final.	<ul style="list-style-type: none">x = variable independiente (cualquier SNA). Si x es RN, 0 < xi ≤ 1t = tipo de la función (C =continua, D =discreta, E =discreta de atributos numéricos) 3n = cantidad de puntos que definen la función	
	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">p = id del parámetro a decrementara = rótulo al que bifurca	
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">op puede ser e, ne, g, ge, l, lea y b pueden ser constantes o SNAsf = rótulo al que bifurca si es falsa	

³Debe 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"> ■ c = cantidad de copias (constante o SNA) ■ r = rótulo al que bifurcan las copias ■ n = identificador de parámetro a utilizar para enumerar el original y las copias 	
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"> ■ f = cantidad de filas ■ c = cantidad de columnas 	
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"> ■ m = id de la matriz ■ f = fila (cualquier SNA) ■ c = columna (cualquier SNA) ■ v = valor a asignar, sumar o restar 	

Bloque	Acción que representa		Parámetros	Valor por defecto
LOGIC v ll	Cambia el estado del <i>logic switch</i> <i>ll</i> por el valor de <i>v</i>		<ul style="list-style-type: none"> ■ <i>v</i> = S (set), R (reset), I (invertido) ■ <i>ll</i> = id de la llave 	
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 <i>c</i> 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 simultáneamente —, la transacción es enviada al rótulo <i>b</i> .		<i>b</i> y <i>c</i> son rótulos.	
INITIAL rll[,v]	Inicializa el estado de la llave <i>rll</i> con el valor <i>v</i> .		<ul style="list-style-type: none"> ■ <i>rll</i> = referencia a una llave lógica ■ <i>v</i> = 1 (set), 0 (reset) 	<i>v</i> = 1(set)
GATE op lógico ⁴ a[,f]	Consulta el estado del <i>logic switch</i> , <i>facility</i> , <i>storage</i> o rótulo <i>a</i> . Si <i>op A</i> es V, continúa en el bloque siguiente. Si es F, bifurca a <i>f</i> . Si no se especifica <i>f</i> , la transacción se bloquea hasta que sea V (<i>delay indicator</i> en ON)		<ul style="list-style-type: none"> ■ <i>a</i> = id del <i>logic switch</i>, <i>facility</i>, <i>storage</i> ■ <i>f</i> = rótulo al que bifurca si la condición es falsa. 	
COUNT op logico ⁴ a,v ₁ ,v ₂	Cuenta la cantidad de objetos desde <i>v₁</i> hasta <i>v₂</i> que cumplen la condición <i>op objeto</i> , y lo almacena en <i>a</i> .		<ul style="list-style-type: none"> ■ <i>a</i> = parámetro que recibe la cantidad de objetos que cumplen la condición ■ <i>v₁</i> = número menor del objeto a consultar ■ <i>v₂</i> = número mayor del objeto a consultar 	

⁴Logic 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 ⁵	a, v ₁ , v ₂ , d, e	Cuenta la cantidad de objetos desde v ₁ hasta v ₂ que cumplen la condición <i>objeto op d</i> , y lo almacena en <i>a</i> .	<ul style="list-style-type: none"> ■ <i>a</i> = parámetro que recibe la cantidad de objetos que cumple la condición ■ <i>v₁</i> = número menor del objeto a consultar ■ <i>v₂</i> = número mayor del objeto a consultar ■ <i>d</i> = valor con el que se compara ■ <i>e</i> = clase de objeto que se selecciona (SNA) 	
COUNT	MIN/MAX	a, v ₁ , v ₂ , , e	Cuenta la cantidad de objetos desde v ₁ hasta v ₂ que son mínimo o máximo, y lo almacena en <i>a</i> .	<ul style="list-style-type: none"> ■ <i>a</i> = parámetro que recibe la cantidad de objetos que son mínimo o máximo ■ <i>v₁</i> = número menor del objeto a consultar ■ <i>v₂</i> = número mayor del objeto a consultar ■ <i>e</i> = clase de objeto que se compara (SNA) 	
	JOIN	g	La transacción activa se une al grupo <i>g</i> “personalmente”.	<i>g</i> = id del grupo.	
	EXAMINE	g, , f	Verifica si la transacción activa pertenece al grupo <i>g</i> . Si pertenece, continúa en el bloque siguiente. Si no, bifurca a <i>f</i> .	<ul style="list-style-type: none"> ■ <i>g</i> = id del grupo. ■ <i>f</i> = rótulo al que bifurca. 	
	REMOVE	g	La transacción activa se elimina al grupo <i>g</i> “personalmente”.	<i>g</i> = id del grupo.	

⁵EQ, L, LE, G, GE, NE

Bloque			Acción que representa	Parámetros	Valor por defecto
REMOVE	[op] ⁶	$g, [c], , [p], [v], [f]$	Intenta eliminar c transacciones del grupo g que cumplen la condición $p \text{ op } v$. Si no puede eliminar todas, bifurca a f .	<ul style="list-style-type: none"> ■ g = id del grupo ■ c = cantidad de transacciones a eliminar ■ p = id del parámetro que se consulta de la transacción ■ v = valor con el que se compara (no se pone si $op = MIN/MAX$) ■ f = rótulo al que bifurca si no pudo eliminar c transacciones 	$op = E$ $c = ALL$
ALTER	[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 $p_2 \text{ op } v_2$ es V, se asigna $p_1 = v_1$. Si no puede alterar c transacciones, bifurca al rótulo f .	<ul style="list-style-type: none"> ■ g = id del grupo ■ c = cantidad de transacciones a eliminar⁸ ■ p_1 = id del parámetro a alterar ■ v_1 = valor a asignar al parámetro p_1 (constante o SNA) ■ p_2 = id del parámetro a consultar ■ v_2 = valor con el que se compara p_2 (constante o SNA) ■ f = rótulo al que bifurca si no pudo alterar c transacciones 	$op = E$
SCAN	[op]	$g, b, [c], p_1, p_2 [, f]$	Busca en el grupo g la primera transacción que cumple la condición $b \text{ op } c$. 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"> ■ OP puede ser $e, g, ge, l, le, max, min, ne$ ■ b = id del parámetro que se consulta ■ c = valor con el que se compara (si OP es min/max, no se pone) ■ p_1 = id del parámetro de la transacción del grupo que se copia ■ p_2 = id del parámetro de la transacción actual que recibe el valor de p_1 	<ul style="list-style-type: none"> ■ $op = E$

⁶E, G, GE, L, LE, MAX, MIN, NE⁷E, G, GE, L, LE, NE⁸ 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 no es encadenada y bifurca al rótulo f .		<ul style="list-style-type: none"> ■ c = id de la cadena (constante o SNA) ■ r = regla por la cual se encadena (FIFO, LIFO, PR, Pn, M1) ■ f = rótulo al que va la primera transacción 	
UNLINK $[op] \quad a, b, c, [d], [e], [f]$	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"> ■ op puede ser E, G, GE, L, LE, NE ■ a = id de la cadena ■ b = rótulo al que se envían las transacciones liberadas ■ c = cantidad de transacciones que se desencadenan⁹ ■ d = id de parámetro de la transacción activa que se consulta ■ e = referencia del parámetro de la transacción pasiva que se consulta ■ f = rótulo al que bifurca la transacción activa si no pudo liberar c transacciones. 	■ $op = E$
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, no 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	

⁹ 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"> ■ <i>a</i> = id de <i>facility</i> ■ <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>) ■ <i>c</i> es obligatorio si <i>b</i> = <i>RE</i> ■ <i>d</i> = parámetro del <i>owner</i> de la <i>facility</i> donde se guarda el tiempo remanente del ADVANCE que estuviera ejecutando. ■ <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>) ■ <i>f</i> es obligatorio si <i>e</i> = <i>RE</i> ■ <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>) ■ si <i>h</i> es usado, <i>g</i> debe ser <i>RE</i> 	
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 testado por transacciones iguales.	
JOIN	g,n	Unir el número <i>n</i> al grupo numérico <i>g</i>	<ul style="list-style-type: none"> ■ <i>g</i>= id del grupo numérico o SNA ■ <i>n</i> =constante o SNA 	
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"> ■ <i>g</i>= id del grupo numérico o SNA ■ <i>n</i> =constante o SNA 	
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"> ■ <i>g</i>= id del grupo numérico o SNA ■ <i>n</i> =constante o SNA 	

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\$FUNCTION

Para referirse a una MATRIX:

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

2. Por número: Mx4(col,fil)