

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>f = función con operadores (cualquier SNA)<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 ₁ ,y ₁ /.../x _n ,y _n	Devuelve un valor cada vez que se la invoca. Define una función de n puntos. Si $x \leq x_1$, devuelve y_1 , si $x \geq x_n$, devuelve y_n . 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, $0 < x_i \leq 1$■ t = tipo de la función (C =continua, D =discreta, E =discreta de atributos numéricos)³■ n = 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 decrementar■ a = rótulo al que bifurca													
TEST	op	a,b[,f]	Si $a \text{ op } b$ 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, le■ a y b pueden ser constantes o SNAs■ f = 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
Llaves lógicas					
LOGIC	v	l	Cambia el estado del <i>logic switch</i> <i>l</i> por el valor de <i>v</i>	<ul style="list-style-type: none">▪ <i>v</i> = S (set), R (reset), I (invertido)▪ <i>l</i> = identificador de la llave	
INITIAL		l[,v]	Inicializa el estado de la llave <i>l</i> con el valor <i>v</i> .	<ul style="list-style-type: none">▪ <i>l</i> = referencia a una llave lógica▪ <i>v</i> = 1 (set), 0 (reset)	<i>v</i> = 1(set)
GATE	op logico 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> = identificador del <i>logic switch</i>, <i>facility</i>, <i>storage</i>▪ <i>f</i> = rótulo al que bifurca si la condición es falsa.	
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.	

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
Conteo (similar a SELECT)					
COUNT	op logico ⁵	p,v ₁ ,v ₂	Cuenta la cantidad de objetos desde v ₁ hasta v ₂ que cumplen la condición op, y lo almacena en el parámetro p.	<ul style="list-style-type: none">▪ p = parámetro que recibe la cantidad de objetos que cumplen la condición▪ v₁ = número menor del objeto a consultar▪ v₂ = número mayor del objeto a consultar	
COUNT	op relacion ⁶	p,v ₁ ,v ₂ ,d,e	Cuenta la cantidad de objetos desde v ₁ hasta v ₂ que cumplen la condición objeto op d, y lo almacena en el parámetro p.	<ul style="list-style-type: none">▪ p = parámetro que recibe la cantidad de objetos que cumple la condición▪ v₁ = número menor del objeto a consultar▪ v₂ = número mayor del objeto a consultar▪ d = valor con el que se compara▪ e = clase de objeto que se selecciona (SNA)	
COUNT	MIN/MAX	p,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 el parámetro p.	<ul style="list-style-type: none">▪ p = parámetro que recibe la cantidad de objetos que son mínimo o máximo▪ v₁ = número menor del objeto a consultar▪ v₂ = número mayor del objeto a consultar▪ e = clase de objeto que se compara (SNA)	

⁵Logic Switch: LS, LR

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

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

⁶EQ, L, LE, G, GE, NE

Bloque		Acción que representa	Parámetros	Valor por defecto
Cadenas de usuarios				
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 = identificador 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	$[op]$ $a, b, c, [d], [e], [f]$	La transacción activa actual desencadena c transacciones de la cadena a tales que se verifique $d o p e$, 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 = identificador de la cadena▪ b = rótulo al que se envían las transacciones liberadas▪ c = cantidad de transacciones que se desencadenan⁷▪ d = identificador 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.	<ul style="list-style-type: none">▪ $op = E$
BUFFER		La transacción actual se coloca como última transacción a mover en la CEC. Detiene a la misma sin que pase el tiempo.		

⁷ c puede ser ALL

Bloque			Acción que representa	Parámetros	Valor por defecto
Grupos de transacciones					
JOIN	g		La transacción se une al grupo g “personalmente”.	g = identificador del grupo.	
REMOVE	g		La transacción se elimina del grupo g “personalmente”.	g = identificador del grupo.	
REMOVE	$[op]^8$	$g, [c], , [p], [v], [f]$	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">▪ g = identificador del grupo▪ c = cantidad de transacciones a eliminar▪ p = identificador 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	<ul style="list-style-type: none">▪ $op = E$▪ $c = ALL$
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">▪ g = identificador del grupo.▪ f = rótulo al que bifurca.	
ALTER	$[op]^9$	$g, c, p_1, v_1, [p_2], [v_2], [f]$	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">▪ g = identificador del grupo▪ c = cantidad de transacciones a eliminar¹⁰▪ p_1 = identificador del parámetro a alterar▪ v_1 = valor a asignar al parámetro p_1 (constante o SNA)▪ p_2 = identificador 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	$op = E$

⁸E, G, GE, L, LE, MAX, MIN, NE⁹E, G, GE, L, LE, NE¹⁰ c puede ser ALL

Bloque			Acción que representa	Parámetros	Valor por defecto
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, NE, G, GE, L, LE, MAX, MIN ■ b = identificador del parámetro que se consulta ■ c = valor con el que se compara (si op es MIN/MAX, no se pone) ■ p_1 = identificador del parámetro de la transacción del grupo que se copia ■ p_2 = identificador del parámetro de la transacción actual que recibe el valor de p_1 	<ul style="list-style-type: none"> ■ $op = E$
Grupo de números					
JOIN		g, n	Unir el número n al grupo numérico g	<ul style="list-style-type: none"> ■ g = identificador del grupo numérico o SNA ■ n = constante o SNA que se une al grupo 	
REMOVE		$g, , n [, , , f]$	Remover al número n del grupo numérico g . Si no pertenece, bifurca a f	<ul style="list-style-type: none"> ■ g = identificador del grupo numérico o SNA ■ n = constante o SNA que se intenta eliminar del grupo ■ f = rótulo al que bifurca si n no pertenece al grupo 	
EXAMINE		g, n, f	Si el número n pertenece al grupo numérico g , continúa en el bloque siguiente. Si no, bifurca a f	<ul style="list-style-type: none"> ■ g = identificador del grupo numérico o SNA ■ n = constante o SNA que se desea encontrar en el grupo ■ f = rótulo al que bifurca si n no pertenece al grupo 	

Bloque	Acción que representa	Parámetros	Valor por defecto
Arrebato de <i>facilities</i>			
PREEMPT <i>f</i>	La transacción actual arrebata la <i>facility f</i> que ha sido tomada por un SEIZE, sin tener en cuenta las prioridades de las transacciones. Si fue tomada por otro PREEMPT, no la arrebata. Interrumpe el ADVANCE del <i>owner</i> de la <i>facility</i> .	<i>f</i> = identificador de <i>facility</i> a intentar arrebatar	
PREEMPT <i>f</i> , PR[, <i>c</i> , <i>d</i> [, RE]]	El arrebato se produce sólo si el <i>owner</i> de la <i>facility f</i> tiene menos prioridad que la transacción actual. Si se especifica RE, el arrebato es definitivo.	<ul style="list-style-type: none"> ■ <i>f</i> = identificador de <i>facility</i> a intentar arrebatar ■ <i>c</i> = rótulo al que bifurca el <i>owner</i> si estaba ejecutando un ADVANCE ■ <i>d</i> = identificador del parámetro del <i>owner</i> donde se almacena el tiempo remanente del <i>owner</i> para terminar el ADVANCE 	
RETURN <i>f</i>	Retorno de la <i>facility f</i> al <i>owner</i> anterior		

Bloque		Acción que representa	Parámetros	Valor por defecto
Rotura de <i>facilities</i>				
FUNAVAIL	a, [b] , [c] , [d] , [e] , f , [g]	Rotura de <i>facility</i> <i>a</i> . El owner 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> = identificador 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 owner 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 <i>ADVANCE</i> que estuviera ejecutando.▪ <i>e</i> = <i>RE</i> (las transacciones interrumpidas la dejan definitivamente) o <i>CO</i> (el owner 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</i> <i>a</i>		
Rotura de <i>storages</i>				
SUNAVAIL	a	Rotura de <i>storage</i> <i>a</i>		
SAVAIL	a	Arreglo de <i>storage</i> <i>a</i>		
Sincronización de transacciones (de la misma familia)				
MATCH	a	Sincronizar el movimiento de transacciones de la misma familia.	<i>a</i> = rótulo para ser testeado por transacciones iguales.	