

QNAP - Introducción

QNAP = Queuing Network Analysis Package

Paquete de software para analizar y simular redes de colas

Está compuesto de dos grandes partes:

1) Un lenguaje de especificación que se usa para

- La descripción de los modelos a analizar
- El control de su resolución

2) Un conjunto de módulos de resolución

- Simulador de eventos discretos
- Un analizador de cadenas de Markov
- Resolutores analíticos exactos
- Resolutores analíticos aproximados

QNAP - Lenguaje Algorítmico

Palabras reservadas:

ALL	NIL	AND	NOT	ANY	OBJECT
BEGIN	OR	DO	REF	ELSE	REPEAT
END	STEP	FALSE	THEN	FOR	TRUE
FORWARD	UNTIL	GENERIC	VAR	GOTO	WATCHED
IF	WHILE	IN	WITH	IS	

Identificadores: 8 caracteres y empiezan por letra

Números: Podrán ser enteros o reales

Cadenas: Cualquier conjunto de caracteres entre comillas dobles

Comentarios: Todo lo que sigue al carácter & hasta el fin de la línea

Separadores: Caracteres blanco y retorno de carro

QNAP - Lenguaje Algorítmico

Expresiones:

Formadas por constantes, variables y/o llamadas a funciones, relacionadas por operadores lógicos, aritméticos y/o por paréntesis

+ Prioridad en la resolución de operadores -

(NOT) () (::, IS, IN) (*, /, AND) (+, -, OR) (=, <>, <, >, <=, >=)**

Sentencias:

Pueden ser simples o compuestas

Sentencia compuesta = Grupo de sentencias simples entre un BEGIN y un END

GOTO etiqueta;

IF expresión THEN sentencia [ELSE sentencia]

WHILE expresión DO sentencia;

FOR variable := lista DO sentencia;

FOR N:=1 STEP 1 UNTIL 100 DO ...

FOR Q:=A, B, C DO ...

WITH identificador_de_objeto DO sentencia;

FOR J:=ALL QUEUE WITH MTHRUPUT(QUEUE) > 0.0 DO

QNAP - Lenguaje Algorítmico

Operaciones de Entrada/Salida:

GET/GETLN ([fichero,] tipo [,longitud])

Funciones que devuelven valores leídos de un fichero con un tipo de dato

Por defecto se leen los datos del fichero FSYSGET

El tipo de dato devuelto puede ser: INTEGER, REAL, BOOLEAN, STRING

La longitud es el número de caracteres a leer

PRINT / WRITE / WRITELN [([fichero,] {dato [,formato]} [,...])]

Por defecto se escriben los datos en el fichero FSYSPRINT=FSYSOUTPUT

El formato se define así:

Entero:N (Donde N define el número de dígitos totales a representar)

Real:N:D (Donde D indica el número de dígitos decimales a representar)

FILASSIGN (fichero, nombre);

OPEN (fichero [,modo]);

CLOSE (fichero [,modo]);

Modo	Acción
1	Leer un fichero existente
2	Crear un fichero nuevo
3	Escribir en un fichero existente

QNAP - Lenguaje de Control

Los comandos de control se escriben entre / / sin espacios de separación entre las barras y el comando

Comandos de control

**/DECLARE/
/STATION/
/CONTROL/
/EXEC/
/TERMINAL/
/REBOOT/
/RESTART/
/END/**

**Declaración de variables e identificadores
Descripción de características de una estación
Especificación de parámetros de control
Bloque de instrucciones de ejecución
Ejecución interactiva
Relanzamiento de la ejecución tras RESTORE
Introducción de un nuevo modelo
Final del fichero QNAP**

Cada uno de los parámetros de control puede aparecer más de una vez dentro de la secuencia de un fichero QNAP

Tiempo de { **Compilación de comandos QNAP (finaliza con un EXEC)
Ejecución de las sentencias del bloque EXEC
Resolución del modelo**

QNAP - Lenguaje de Control - /DECLARE/

Comando usado para declarar los datos y variables del programa QNAP

Datos de tipo escalar

Tipo de Dato	Ejemplo de declaración
Entero	INTEGER L1=10;
Real	REAL R=5.2E4;
Lógico	BOOLEAN Lbs;
Cadena	STRING D;

Datos de tipo array

Consisten en una colección de elementos del mismo tipo

No están permitidos los arrays de arrays

REAL C(5); & Array (vector) de 5 reales

INTEGER D(2,3); & Array (matriz) de 6 enteros

Datos de tipo referencia

Permiten referenciar a objetos. Equivalen a los punteros en otros lenguajes

REF tipo-objeto identificador;

QNAP - Lenguaje de Control - /DECLARE/

Datos de tipo objeto

Son datos complejos que puede definir el usuario

```
/DECLARE/  
  OBJECT CENTRO (PROB, NLINES);  
    & Declaración de atributos  
    INTEGER NLINES;          & Número de líneas  
    REAL PROB (NLINES);      & Probabilidades  
    REAL TLINE;              & Tiempo de servicio promedio  
END;
```

Mas adelante puede escribirse:

```
/DECLARE/  
  CENTRO ((0.5, 0.2, 0.2, 0.1), 4) CT;
```

Declaración de un objeto

```
[tipo-obj] OBJECT Identificador [(parámetros)];  
  declaración de atributos;  
END;
```

Ejemplo

```
/DECLARE/  
  CUSTOMER OBJECT PACKET;  
  INTEGER BYTES;  
  STRING HEADER;  
  REF CLASS CIRCUIT;  
END;
```

QNAP - Lenguaje de Control - /DECLARE/

Etiquetas Son identificadores empleados para marcar el destino de los saltos
LABEL identificador

Procedimientos

Subrutinas de código escrito por el usuario

Declaración habitual:

```
PROCEDURE Nombre [(parámetros)];  
[declaración de parámetros];  
[declaración de variables locales];  
BEGIN  
    Sentencias;  
END;
```

Transmisión de parámetros

- Por valor
- Por referencia
(Precedidos por la palabra VAR)

/DECLARE/

```
QUEUE Q;  
REAL SUM=0.0; T=3.0;  
PROCEDURE DELAY(DELTA, D);  
    VAR REAL DELTA;  
    REAL D;  
    BEGIN  
        D:=2*D;  
        CST(D);  
        DELTA:= DELTA+D;  
    END;
```

/STATION/

```
NAME=Q;  
SERVICE=DELAY(SUM,T);
```


QNAP - Lenguaje de Control - /DECLARE/

Funciones

Son similares a los procedimientos y se diferencian en:

- 1) La palabra reservada es FUNCTION en lugar de PROCEDURE
- 2) La función va precedida del tipo de resultado que devuelve
- 3) El valor devuelto se asigna con la variable predefinida RESULT

/DECLARE/

```
BOOLEAN FUNCTION ISREADY(RQ);  
  REF QUEUE RQ;  
  REF CUSTOMER RC;  
  BEGIN  
    RC:=RQ.FIRST;  
    IF(RC=NIL)  
      THEN RESULT:=TRUE;  
      ELSE RESULT:=NOT RC.BLOQUED;  
  END;
```

QNAP - Lenguaje de Control - /DECLARE/

Objetos predefinidos (predeclarados) en QNAP

QUEUE.- Cola	TIMER.- Temporizador
CUSTOMER.- Cliente	EXCEPTION.- Código a ejecutar al producirse una excepción (en sistemas UNIX)
CLASS.- Clase	
FLAG.- Flag	FILE.- Fichero

Estos objetos tienen declarados una serie de atributos

Pero se pueden añadir nuevos atributos, simplemente declarándolos

```
QUEUE INTEGER H;
```

Los objetos pueden formar parte de otros objetos

```
OBJECT NODE(N,T);  
  INTEGER N;  
  QUEUE CPU;  
  QUEUE DISK(N);  
  REAL T;  
END;
```

QNAP - Lenguaje de Control - /DECLARE/

Atributos de los objetos del tipo COLA

CAPACITY.- Informa sobre la capacidad de la cola
[queue.] capacity [(clase)];

MULT.- Informa sobre la multiplicidad o número de servidores en la cola
Devuelve el valor (-1) en el caso de que sea un servidor infinito

FIRST.- Referencia al primer cliente de la cola

LAST.- Referencia al último cliente de la cola

NB.- Número de clientes en la cola (esperando y recibiendo servicio)

NBIN.- Número de clientes que han entrado en la cola hasta el momento

NBINSERV.- Número de clientes que están recibiendo servicio

NBOUT.- Número de clientes que han salido de la cola hasta el momento

VALUE.- Número de servidores libres

Atributos de los objetos del tipo FLAG

STATE.- Variable booleana (TRUE si el flag está SET, FALSE si está RESET)

LIST.- Referencia al último cliente esperando por el flag

QNAP - Lenguaje de Control - /DECLARE/

Atributos de los objetos del tipo CLIENTE

- ACTIVETIME.-** Tiempo durante el que el cliente está activo
- BLOCKED.-** Estado actual de bloqueo de un cliente (sí o no)
- BLOCKTIME.-** Tiempo que el cliente ha estado bloqueado
- CHANGEDATE.-** Fecha de la última modificación del estado de un cliente
(activo, esperando, bloqueado)
- ENTERDATE.-** Fecha de llegada a la estación actual
- RESPTIME.-** Tiempo empleado por el cliente en una estación
- STARTED.-** Fecha de comienzo de servicio a un cliente
- CCLASS.-** Referencia a la clase del cliente
- CPRIOR.-** Nivel de prioridad del cliente
- CQUEUE.-** Referencia la estación que contiene al cliente
- FATHER.-** Referencia al padre del cliente
- NEXT.-** Referencia al cliente siguiente en la cola
- PREVIOUS.-** Referencia al cliente previo en la cola
- SON.-** Referencia al último cliente creado por el cliente

QNAP - Lenguaje de Control - /STATION/

Comando usado para describir las estaciones de servicio del modelo (incluyendo la definición inicial y las sucesivas alteraciones)

Parámetros: NAME, TYPE, SCHED, PRIOR, QUANTUM, INIT, SERVICE, TRANSIT, SPLIT, MATCH, FISSION, FUSION, RATE, CAPACITY, REJECT, COPY

NAME.- Define (asigna nombre) a la cola o colas (estaciones)
Debe ser el primer parámetro del comando

```
/DECLARE/  
  QUEUE CPU, DK(10);
```

```
/STATION/  
  NAME=CPU;  
  ...
```

```
/STATION/  
  NAME=DK(1 STEP 1 UNTIL 5);  
  TRANSIT=CPU;  
  SERVICE=EXP(0.1);
```

QNAP - Lenguaje de Control - /STATION/

TYPE.- Define el tipo de estación

Se utiliza: TYPE=tipo;

Los tipos pueden tomar los valores siguientes:

- SERVER	Estación de tipo servidor
SINGLE	1 servidor
MULTIPLE(n)	n servidores
INFINITE	infinitos servidores
- RESOURCE	Estación de tipo recurso
SINGLE	Recurso no compartido
MULTIPLE(n)	Recurso compartible por n clientes
INFINITE	Recurso que está siempre disponible
- SEMAPHORE	Estación de tipo semáforo
SINGLE	Valor inicial=1
MULTIPLE(n)	Valor inicial=n ($n \geq 0$)
- SOURCE	Estación fuente (generadora de clientes)
- MULTIPLE(n)	Estación de tipo servidor con n servidores

La ausencia del tipo al definir una estación equivale a TYPE=SERVER, SINGLE

QNAP - Lenguaje de Control - /STATION/

SCHED.- Define la disciplina de servicio en la estación
Se utiliza: SCHED=valor;

Los valores más frecuentes que suele tomar son los siguientes:

- **FIFO.-** Los clientes son servidos según el orden de llegada
- **LIFO ó FILO.-** Los clientes son servidos en orden inverso al de llegada
- **PRIOR.-** Los clientes se ordenan en la cola según su prioridad relativa
- **PREEMPT.-** El cliente que está recibiendo servicio en un recurso es desalojado del recurso por la llegada de un cliente de más prioridad. El cliente desalojado continúa recibiendo servicio cuando vuelve a acceder al recurso
- **QUANTUM.-** Se asigna un servidor a los clientes por periodos de longitud fija (cuanto) definidos por un número real
- **PS.-** Se comparte el servidor (procesador) por igual entre todos los clientes a servir

La ausencia de un valor al definir una estación equivale a SCHED=FIFO

QNAP - Lenguaje de Control - /STATION/

PRIOR.- Fija el nivel de prioridad de un cliente que entra en la estación

Puede ser diferente para cada clase de clientes

Sin parámetros se aplica a todas las clases

Se utiliza: PRIOR [(lista de clases)]=entero; (mayor entero = mayor prioridad)

```
/DECLARE/ QUEUE UC, UK;  
            INTEGER PX;  
            CLASS X,Y,Z;
```

```
/STATION/ NAME=UC;  
          PRIOR(X,Y)=PX; & Para las clases X e Y  
          PRIOR=2;      & Para el resto de clases
```

```
/STATION/ NAME=UK;  
          SCHED=PRIOR, PREEMPT;  
          PRIOR(Y)=1;  
          PRIOR(Z)=3;  
          & La clase X mantiene la prioridad que tuviera
```


QNAP - Lenguaje de Control - /STATION/

QUANTUM.- Especifica la asignación del cuanto en cada estación

Sólo se usa si se ha elegido QUANTUM en el parámetro SCHED

Si no se especifica una clase se aplica el valor para todas las clases

Se utiliza: QUANTUM [(lista de clases)]=real;

```
/DECLARE/ QUEUE UC;  
            REAL QT;  
            CLASS X,Y,Z;
```

```
/STATION/  NAME=UC;  
            SCHED=QUANTUM(0.3);  
            QUANTUM(X)=QT;
```

```
/DECLARE/  QUEUE A,B;  
            INTEGER N;  
            CLASS X,Y;
```

```
/STATION/  NAME=A;  
            INIT(X)=2;  
            INIT(Y)=1;  
/STATION/  NAME=B;  
            INIT=N;
```

INIT.- Especifica el número de clientes de cada clase que están en la estación en el instante considerado como inicial

Si no se especifica una clase se aplica el valor para todas las clases

Las estaciones Resource y Semaphore no deben tener clientes inicialmente

Se utiliza: INIT [(lista de clases)]=entero;

QNAP - Lenguaje de Control - /STATION/

SERVICE.- Especifica el servicio requerido por los clientes en la estación
Puede ser distinto para cada clase y se ignora para Resource y Semaphore
Se utiliza: SERVICE [(lista de clases)]=sentencia;

Resolución analítica: el servicio sólo contiene una sentencia

Simulación: el servicio puede contener múltiples sentencias entre BEGIN y END

- Modelización de operaciones realizadas por el servicio:

 - Usar las construcciones del lenguaje algorítmico

- Modelización del tiempo consumido por el servicio:

CST(real);	Constante
EXP(real);	Exponencial(media)
HEXP(real1, real2);	Hiperexponencial(media, varianza/media^2)
ERLANG(real, entero);	Erlang(media, nº exp a sumar)
UNIFORM(real1, real2);	Uniforme(extremo Izq, extremo Dch)
RINT(entero1, entero2);	Uniforme entre valores enteros
COX;	Ley de Cox
NORMAL(real1, real2);	Normal(media, desviación)

QNAP - Lenguaje de Control - /STATION/

TRANSIT.- Describe el encaminamiento de los clientes al terminar el servicio usando

Una o varias estaciones de destino

Una o varias clases de destino

Una o varias probabilidades de transición (relativas o absolutas)

Se utiliza: TRANSIT [(lista-clases)] {lista-colas [,lista-clases], probabilidad};

Destrucción de clientes: Enviándolos a la **cola OUT** (en sistemas abiertos)

Generación de clientes: Usando una **estación fuente** que les asigna una clase

```
/DECLARE/  
  QUEUE A,B,C;  
  CLASS X,Y,Z;  
  REAL PA,PB;  
/STATION/  
  NAME=A;  
  SERVICE=EXP(5);  
  TRANSIT=B;  
  TRANSIT(X)=A,PA, B,Y,PB, C,Z;  
  TRANSIT(Y)=A, Z, 1, C, Y, 2;
```

```
/DECLARE/  
  QUEUE A,B,C;  
  REAL PROB(3);  
/STATION/  
  NAME=CPU;  
  TRANSIT=DISK (1 STEP 1 UNTIL 3), PROB, OUT;
```

```
/DECLARE/ QUEUE A,B,C;  
/STATION/  NAME=A;  
           SERVICE=EXP(5);  
           TRANSIT=A,0.5, B,0.3, C;
```

QNAP - Lenguaje de Control - /STATION/

CAPACITY.- Especifica el número máximo de clientes permitidos en una estación

La capacidad puede ser diferente para cada clase

Si no se especifica una clase la capacidad se refiere a toda la estación

Por defecto la capacidad se supone infinita

Se utiliza: CAPACITY [(lista-clases)]=entero;

COPY.- Define estaciones idénticas copiando los parámetros de una estación a otra

Se utiliza: COPY=cola;

```
/DECLARE/ QUEUE CPU, DK0, DK1, DK2;  
            CLASS X,Y;  
/STATION/  NAME = DK0;  
            SERVICE = EXP(10);  
            SCHED = LIFO, PREEMPT;  
            PRIOR(X) = 2;  
            PRIOR(Y) = 1;  
            CAPACITY = 10;  
            TRANSIT = CPU;  
/STATION/  NAME = DK1;  
            COPY = DK0;
```

QNAP - Lenguaje de Control - /CONTROL/

Comando usado para modificar los valores de ejecución tomados por defecto

Parámetros: CLASS, MARGINAL, OPTION, UNIT, TSTART, TMAS, RANDOM, ACCURACY, TEST, PERIOD, TRACE, NMAX, ENTRY, EXIT, ALIAS

CLASS.- Especifica para qué clases se obtienen valores estadísticos
Se utiliza: CLASS=lista de colas;

CLASS=ALL QUEUE; & Para todas las colas y clases
CLASS=NIL; & Para ninguna (defecto)

MARGINAL.- Establece las probabilidades marginales a calcular para cada cola
Probs Marginales: Prob de tener 0, 1, 2 ... clientes en una cola
Por defecto se calculan las probabilidades hasta el nivel 5
Se utiliza: MARGINAL={sublista de colas [,entero]};

QNAP - Lenguaje de Control - /CONTROL/

OPTION.- Controla los resultados que imprime QNAP después de la ejecución
Se utiliza: OPTION=opción, opción, ... ;

Las opciones posibles son (subrayado = defecto):

<u>SOURCE</u>	& Se muestra el programa fuente
NSOURCE	
<u>RESULT</u>	& Aparecen los resultados finales
NRESULT	
TRACE	& Aparecen los valores de traza
<u>NTRACE</u>	
DEBUG	& Resultados del depurador de línea
<u>NDEBUG</u>	
VERIF	& Valores de pruebas en ejecución
<u>NVERIF</u>	

RANDOM.- Cambia la semilla del generador de números aleatorios (defecto=413)
Se utiliza: RANDOM=entero; (una constante o una expresión)

QNAP - Lenguaje de Control - /CONTROL/

UNIT.- Permite especificar la asignación de unidades de entrada/salida
Permite redireccionar la entrada/salida estándar a ficheros
Se utiliza: UNIT=unidad(fichero);

Las unidades son las siguientes:

Ejemplos:

OUTPUT	Salida estándar
PRINT	Salida de impresora estándar
INPUT	Entrada estándar
GET	Entrada interactiva de datos
LIBR	Entrada para librerías
TRACE	Salida de resultados de traza

```
FILEASSIGN(F,"nombre.dat");  
OPEN(F,1);  
UNIT=GET(F);
```

TSTART.- Instante de comienzo de la toma de medidas en una simulación para calcular estadísticas (defecto=0.0)
Se utiliza: TSTART=real;

TMAX.- Duración máxima de una simulación (defecto=0.0)
Se utiliza: TMAX=real; (una constante o una expresión)

QNAP - Lenguaje de Control - /CONTROL/

ACCURACY.- Muestra el intervalo de confianza de un trabajo
Se utiliza: ACCURACY={sublista de colas [,sublista de clases]}

ACCURACY=NIL; & No se precisan intervalos de confianza (defecto)
ACCURACY=ALL QUEUE, ALL CLASS; & Para todas las colas y clases

TEST.- Especifica una sentencia que se ejecutará durante la simulación al final de cada intervalo definido por el parámetro PERIOD
Puede ser una sentencia única o un bloque algorítmico
Se utiliza: TEST=sentencia; (el defecto es TEST=;)

PERIOD.- Define el intervalo de activación de la sentencia de TEST
Se utiliza: PERIOD=real;

```
/DECLARE/  QUEUE A, B, C;  
/CONTROL/  PERIOD = 200;  
           TMAX = 2000.0;  
           TEST = IF A.NBOUT > 500 THEN STOP;  
/EXEC/     SIMUL;
```


QNAP - Lenguaje de Control - /CONTROL/

TRACE.- Define los instantes inicial y final de trazado durante una simulación

El valor por defecto es TRACE=0.0, 0.0;

Si no se especifica el segundo real se toma el valor TMAX

Cadena indica el número de columnas del formato de salida "L80" ó "L132"

Se utiliza: TRACE=real [,real] [,cadena];

NMAX.- Asigna el número máximo de clases utilizable

Se utiliza: NMAX=entero; (defecto=20)

ALIAS.- Crea nuevos nombres para palabras clave ya existentes

El primer identificador es el alias dado al segundo identificador

Se utiliza: ALIAS={(identificador, identificador)} [, ...];

/CONTROL/ ALIAS=(NOMBRE, NAME), (PLANIFICACION, SCHED);

QNAP - Lenguaje de Control - /CONTROL/

ENTRY.- Define una sentencia que se ejecuta antes de la resolución
Se utiliza: ENTRY=sentencia; (generalmente para inicializar variables)

EXIT.- Define una sentencia que se ejecuta después de la resolución
Se utiliza: EXIT=sentencia; (generalmente para presentar resultados)

```
/DECLARE/  QUEUE A, B, C;
/CONTROL/  ENTRY=BEGIN
                PRINT("MA, MB, MC", MA, MB, MC);
                PB:=MB; PB:=PB / (MA+MB+MC);
                PC:=MC; PC:=PC / (MA+MB+MC);
            END;
        EXIT=BEGIN
                LRESPONS(N):=MRESPONS(C);
                PRINT(MRESPONS(A)+MRESPONS(B));
            END;
/EXEC/  BEGIN
        MA:=10; MB:=20; MC:=30; SOLVE;
        END;
```

QNAP - Lenguaje de Control - /EXEC/

Comando usado para describir la parte dedicada a la resolución del modelo

Los métodos utilizables son los siguientes:

SOLVE [("keyword")];

Para resolución analítica (La keyword especifica el método)

Si se omite, QNAP selecciona el método más apropiado

"CONVOL"	Algoritmos de convolución
"MVA"	Algoritmos de análisis del valor medio
"MVANCA"	Algoritmos MVA + convolución normalizada
"PRIORPR"	Algoritmos para planificación preemtiva con prioridades
"HEURSNC"	Algoritmos heurísticos
"ITERATIV"	Algoritmos iterativos
"DIFFU"	Algoritmos de difusión

MARKOV;

Para la resolución de cadenas de Markov

SIMUL;

Para realizar una simulación de eventos discretos

QNAP - Funciones (1): Matemáticas

ABS(entero/real);	Valor absoluto de un entero o un real
ACOS(real);	Función trigonométrica Arco-Coseno; Dominio=[-1,1]
ASIN(real);	Función trigonométrica Arco-Seno; Dominio=[-1,1]
ATAN(real);	Función trigonométrica Arco-Tangente
COS(real);	Función trigonométrica Coseno; Argumento en radianes
EXPO(real);	Función exponencial
FIX(real);	Parte entera de un real, devuelto en formato real
INTREAL(real);	Parte entera de un real, devuelto en formato entero
INTROUND(real);	Entero más próximo a un real
LOG(real);	Función logaritmo natural o neperiano
LOG10(real);	Función logaritmo decimal
MAX(ent1/real1, ...);	Máximo de una lista de valores
MIN(ent1/real1, ...);	Mínimo de una lista de valores
MOD(ent1, ent2);	Resto de la división de dos enteros
REALINT(entero);	Conversión de un entero a formato real
SIN(real);	Función trigonométrica Seno; Argumento en radianes
SQRT(real);	Raíz cuadrada; Argumento ≥ 0
TAN(real);	Función trigonométrica Tangente; Argumento en radianes
DISCRETE(lista-val, lista-prob [,entero]);	Devuelve un valor aleatorio de una lista

QNAP - Funciones (2): Manejo de objetos

NEW(type [,list-par]);

Crea dinámicamente un nuevo objeto del tipo **type** con los parámetros **list-par**

Devuelve una referencia al nuevo objeto creado

Por defecto el objeto hijo tiene la misma clase y prioridad que el padre

Un cliente creado con NEW debe ser transferido a una cola usando TRANSIT

DISPOSE(ref.objeto);

Borra objetos creados dinámicamente con NEW

Un objeto creado en /DECLARE/ no puede borrarse

Los clientes se borran con una transición al dispositivo **OUT**

REFSON(cliente, ent);

Devuelve la referencia al cliente hijo definido por **cliente**

El número de hijo buscado viene dado por el segundo argumento **ent**

Los hijos se numeran de más viejo (1) a más joven

TYPENAME(ref.objeto [,ent]);

Devuelve el tipo del objeto especificado

QNAP - Funciones (3): Sincronización

BLOCK (cola1, cola2, ...);	Bloquea colas
UNBLOCK (cola1, cola2, ...);	Desbloquea colas
ISBLOCK (cola);	¿Está bloqueada la cola?: True o False
FREE (customer, [,cola/flag]);	Desbloquea un cliente
MOVE (cola1,cola2);	Mueve el primer cliente de una cola a otra
PRIOR ([customer,] integer);	Actualiza la prioridad de un cliente
SET (flag);	Pone el flag en estado alto
RESET (flag);	Pone el flag en estado bajo
WAIT ([cliente,] flag);	Esperar por un flag
WAITAND ([cliente,] flag1,flag2,...);	Espera por todos los flags indicados
WAITOR ([cliente] flag1,flag2,...);	Espera por cualquiera de los flags indicados
TRANSIT ([cliente,] cola, [clase] [,entero]);	Define el tránsito de los clientes

QNAP - Funciones (4): Presentación resultados

CBLOCKED (cola [,clase]);	Intervalo de confianza del tiempo de bloqueo
CBUSYPCT (cola [,clase]);	Intervalo de confianza del porcentaje de ocupación
CCUSTNB (cola [,clase]);	Intervalo de confianza del número medio de clientes
CRESPONSE (cola [,clase]);	Intervalo de confianza del tiempo de respuesta
CSERVICE (cola [,clase]);	Intervalo de confianza del tiempo medio de servicio
CUSTNB (cola [,clase]);	Número medio de clientes actualmente en la cola
MAXCUSTNB (cola [,clase]);	Máximo número de clientes en la cola
MBLOCKED (cola [,clase]);	Tiempo medio de bloqueo de la cola
<u>MBUSYPCT</u> (cola [,clase]);	Porcentaje medio de ocupación de la cola
<u>MCUSTNB</u> (cola [,clase]);	Número medio de clientes en la cola
<u>MRESPONSE</u> (cola [,clase]);	Tiempo medio de respuesta de la cola
<u>MSERVICE</u> (cola [,clase]);	Tiempo medio de servicio de la cola
<u>MTHRUPUT</u> (cola [,clase]);	Productividad media de la cola
PCUSTNB (ent, cola [,clase]);	Probabilidad de tener <i>ent</i> clientes en la cola
PMXCUSTNB (cola [,clase]);	Número máximo de clientes en la cola (último período)
SERVNB (cola [,clase]);	Número de clientes servidos por la cola
SONNB (cliente);	Número de hijos del cliente
VCUSTNB (cola [,clase]);	Varianza del número medio de clientes en la cola
VRESPONSE (cola [,clase]);	Varianza del tiempo medio de respuesta de la cola