UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID

SOFTWARE DE SIMULACIÓN

GPSS WORLD

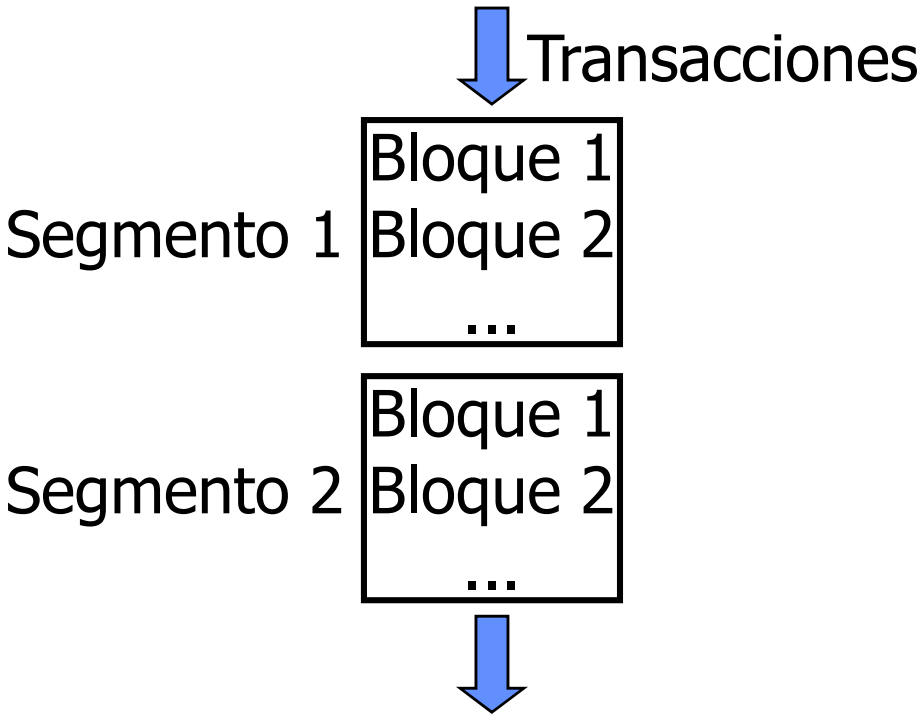
CONTENIDOS

- **Parte general**
  - Segmentos y bloques
  - Creación de transacciones
  - Finalización la simulación
  - Comentarios en el modelo
  - Tiempo de utilización
  - Almacenamiento de resultados
  - Recursos unitarios
  - Colas de espera
  - Tablas de tiempos de espera
  - Recursos de capacidad múltiple
  - Direccionamiento de transacciones
  - Creación generalizada
  - Atributos numéricos
  - Direccionamiento condicional
  - Operadores matemáticos y lógicos

- **Opciones adicionales**
  - Prioridades
  - Interrupciones
  - Funciones aleatorias
  - Variables
  - Funciones adicionales
  - Guardar valores
  - Histogramas
  - Parámetros
  - Seleccionar servidor
  - Loop
  - Separación y unión
  - Obtención de varias muestras

PARTE GENERAL

- El código de simulación se edita en una ventana de texto
- Los componentes que evolucionan en el sistema se denominan TRANSACCIONES (clientes, personas, objetos, vehículos,...)
- El ciclo de vida de las transacciones se describe en lo que se denomina SEGMENTO (la evolución del tráfico en cada vía de un cruce es un segmento distinto)
- Los segmentos están integrados por BLOQUES o COMANDOS (código)



SEGMENTOS Y BLOQUES

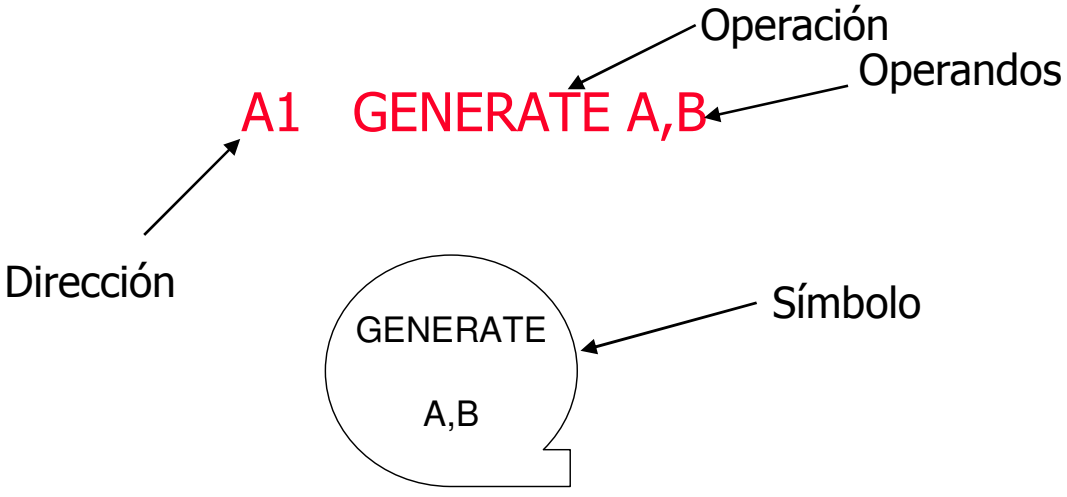
- Cada BLOQUE refleja una fase del ciclo de vida de la transacción dentro del SEGMENTO (ej. fase puede ser 'estar en la ventanilla')

- La estructura general de un bloque :
  - Campo de dirección (opcional)
  - Campo de operación
  - Campo de operandos

- Cada segmento se puede representar mediante un diagrama de bloques

- Cada bloque puede identificarse mediante un símbolo

- Ejemplo: BLOQUE GENERATE



CREACIÓN DE TRANSACCIONES
GENERATE A,B

- Genera una transacción con una distribución uniforme distribuida entre

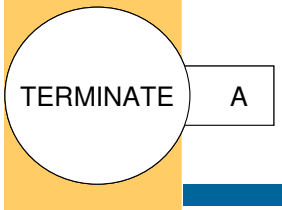
  [A-B, A+B]

- Los operandos A y B no pueden ser negativos (A es como la media y B como la desviación)

- Además el operando A ≥ B

- Si B=0 implica que la generación se realiza a intervalos constantes (B puede omitirse)

- Valor por omisión de A y B es 0



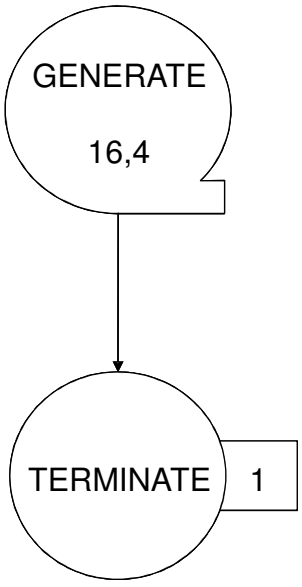
FINALIZACIÓN DE TRANSACCIONES
TERMINATE, START

- TERMINATE se coloca siempre con posterioridad al bloque GENERATE y elimina transacciones del sistema

- El bloque START se coloca a continuación de TERMINATE cuando se quiere limitar el número de transacciones que han completado la simulación

- El bloque START con su operando crea un contador del cual TERMINATE va descontando una cantidad cada vez que pasa una transacción (START indica el valor inicial del contador)

- Ejemplo:




```
GENERATE    16,4
TERMINATE   1
START       30
```

Cada vez que pasa una transacción se descuenta 1 a 30

Se generan **30** transacciones con intervalos de tiempo distribuidos uniformemente **entre 12 y 20**

COMENTARIOS DEL CÓDIGO

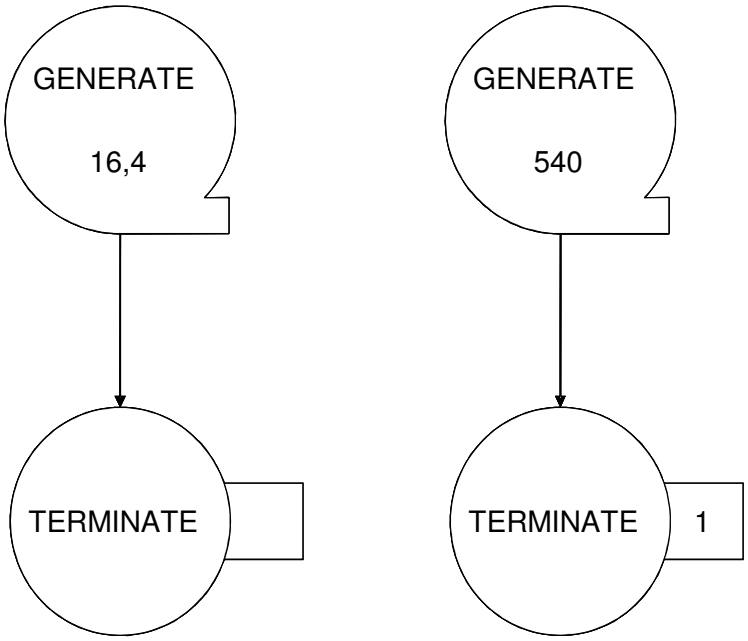
- Los comentarios pueden introducirse de dos formas:
  - Poniendo un * en la columna 1
  - Al final de un bloque con un símbolo ;

- Ejemplo:

Una sucursal bancaria abre 9 horas.

Los clientes llegan uniformemente entre 12 y 20 minutos

```
*  Segmento de generación de clientes
GENERATE 16,4   ; Llegadas de clientes
TERMINATE
*  Segmento de parada
GENERATE 540 ; Cierre después de 9 horas
TERMINATE 1
START     1
```



ADVANCE
A,B

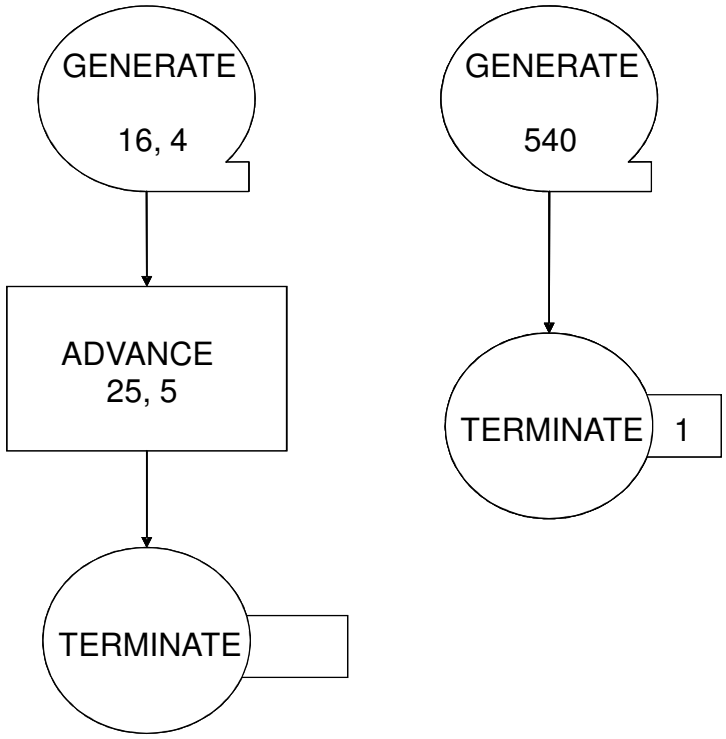
TIEMPO DE ESTANCIA
ADVANCE A,B

• Permite a una transacción permanecer en el segmento un tiempo uniformemente distribuido en el intervalo [A-B, A+B] antes de seguir avanzando

• Ejemplo (continuación):

Cada cliente de la sucursal permanece un tiempo uniformemente distribuido entre 20 y 30 minutos aunque todos los que estén en la sucursal a la hora del cierre se marchan sin ser atendidos

```
GENERATE   16,4
ADVANCE    25,5
TERMINATE

GENERATE   540
TERMINATE  1
START      1
```



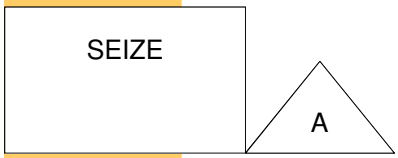
RESULTADOS Y SU ALMACENAMIENTO

- Los resultados del ejemplo anterior se indican por pantalla

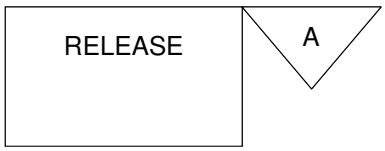
```
START TIME          END TIME    BLOCKS    FACILITIES    STORAGES
    0.000             540.000       5           0             0

                LOC     BLOCK TYPE    ENTRY COUNT    CURRENT COUNT
                1       GENERATE      33             0
                2       ADVANANCE     33             1
                3       TERMINATE     32             0
                4       GENERATE      1              0
                5       TERMINATE     1              0
```

- Los ficheros resultantes de la simulación:
  - Código de simulación: *.gps
  - Resultado de la simulación: *.gpr
  - Bitácora de la ejecución: *.sim

- Los ficheros se guardan mediante las opciones "Save" o "Save as" en el Menú (File)



RECURSO UNITARIO
SEIZE A– RELEASE A

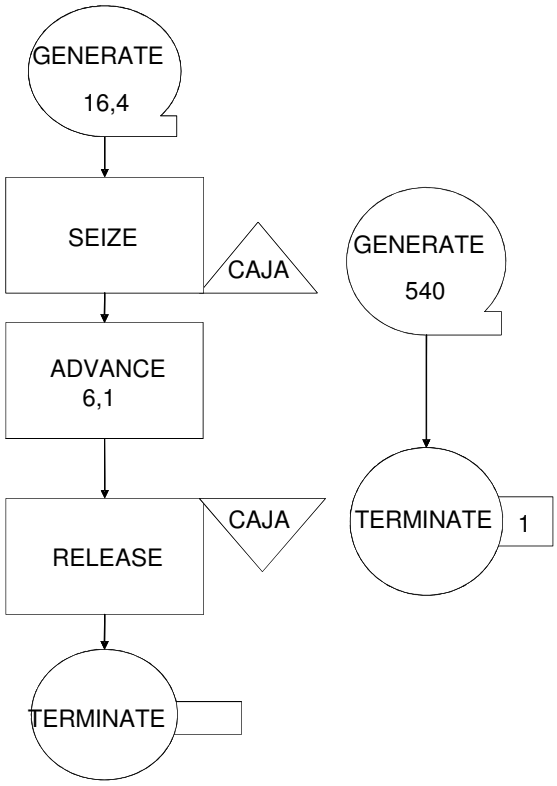


- SEIZE permite a una transacción utilizar un recurso (servidor) que sólo puede atender a una transacción a la vez (se denomina "Facility")
- RELEASE libera el recurso de la transacción
- El operando A de SEIZE y RELEASE identifica el recurso que es utilizado y liberado
- El nombre del recurso puede contener cualquier número de caracteres siempre que el 1º sea una letra
- Si el recurso está ocioso, la transacción puede tomarlo y continuar al bloque siguiente sin que otra transacción pueda tomarlo hasta que no se libere con el bloque RELEASE
- Si el recurso está ocupado, la transacción espera y forma una cola con disciplina FIFO (sin embargo no se proporciona resultados de esta cola)
- Es importante no olvidar el bloque RELEASE ya que el recurso siempre estaría ocupado para las transacciones posteriores a la primera

SEIZE – RELEASE (Ejemplo)

- Ejemplo (continuación)

La sucursal bancaria dispone de un empleado en la CAJA el cual atiende durante un tiempo que se distribuye uniformemente entre 5 y 7 min.




```
GENERATE   16,4
SEIZE      CAJA
ADVANCE     6,1
RELEASE    CAJA
TERMINATE

GENERATE   540
TERMINATE   1
START       1
```

33 entradas a la caja

La caja ha sido utilizada el 36.4% del tiempo

Tiempo medio de servicio

LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1	GENERATE	33	0	0
2	SEIZE	33	0	0
3	ADVANCE	33	0	
4	RELEASE	33	0	
5	**TERMINATE**	**33**	0	
6	GENERATE	1	0	
7	**TERMINATE**	**1**	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.
CAJA	**33**	**0.364**	**5.958**	**1**

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT
35	0	542.978	35	0	1
36	0	1080.000	36	0	60

INFORMACIÓN CON QUEUE Y DEPART

- Se puede obtener información:
  - colas de espera para un recurso
  - transacciones entre dos bloques de un segmento
- Para colas: Se han de situar a ambos lados del bloque que asigna recurso
- Entre dos bloques: antes del bloque inicial y después del bloque final

```
GENERATE   16,4
QUEUE      ESPERA_CAJA ; Cola de espera
SEIZE      CAJA
DEPART     ESPERA_CAJA ; Salida de la cola
ADVANCE    6,1
RELEASE    CAJA
TERMINATE

GENERATE   540
TERMINATE  1
START      1
```

1 como máximo en la cola

34 que han entrado

33 que no han esperado nada

Resultados:

```
FACILITY            ENTRIES   UTIL.    AVE. TIME AVAIL.
CAJA                   34     0.364        5.958   1

QUEUE                 MAX CONT. ENTRY ENTRY(0)
ESPERA_CAJA             1    0     34     33
```

TABLAS DE COLAS
QTABLE A,B,C,D

- Obtiene una tabla de frecuencias del tiempo de espera en una cola
- Los operandos tienen la siguiente función:
  - A : nombre de la cola
  - B : límite superior del primer intervalo (0 si se quieren contabilizar los que no esperan)
  - C : Amplitud de cada intervalo
  - D : Número total de intervalos

Ponerlo siempre al inicio del código

```
FREC        QTABLE ESPERA_CAJA,0,10,20
GENERATE    16,4
QUEUE       ESPERA_CAJA  ;Cola de espera
SEIZE       CAJA
DEPART      ESPERA_CAJA ;Salida de la cola
ADVANCE     20,5
RELEASE     CAJA
TERMINATE
GENERATE     540
TERMINATE    1
START        1
```

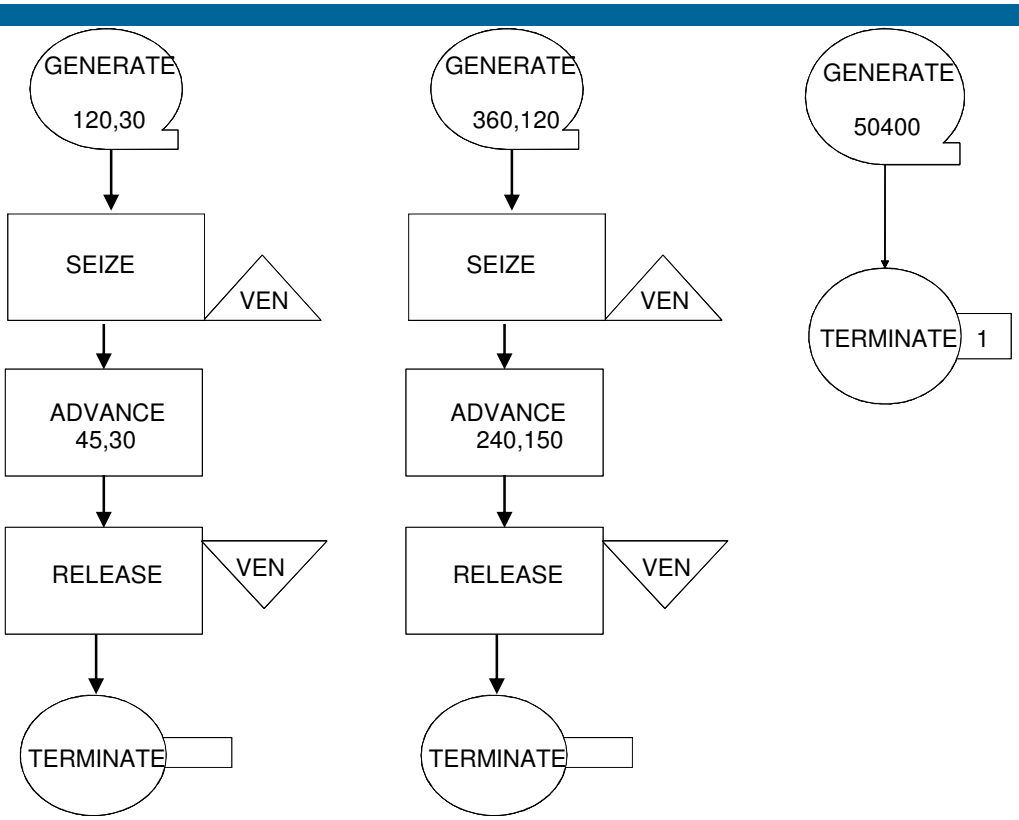
Importante para decir que es tabla de frecuencias

RANGE			RETRY	FRECUENCY	CUM %
	-	0.000	1		3.85
0.000	-	10.000	1		7.69
10.000	-	20.000	4		23.08
20.000	-	30.000	1		26.92
30.000	-	40.000	5		46.15
40.000	-	50.000	1		50.00
50.000	-	60.000	1		53.85
60.000	-	70.000	3		65.38
70.000	-	80.000	1		69.23
80.000	-	90.000	2		76.92
90.000	-	100.000	1		80.77
100.000	-	110.000	2		88.46
110.000	-	120.000	1		92.31
120.000	-	130.000	2		100.00

Ejemplo con diferentes usuarios y el mismo recurso

- Una tienda atiende a clientes que acuden a comprar el periódico o a comprar alimentos y que está atendida por un solo dependiente
- Los compradores de periódicos llegan según la ley U[90,150] (tiempo en segundos) y requieren un servicio de duración aleatoria U[15,75]
- Para los compradores de alimentos estas distribuciones son respectivamente U[240,480] y U[90,390]
- La tienda abre durante 14 horas (50.400 segundos).

Ejemplo con diferentes usuarios y el mismo recurso (Continuación)




```
*Inicio segmento compradores de periódicos
GENERATE   120,30
QUEUE      cola_ven
SEIZE      vendedor
DEPART     cola_ven
ADVANCE    45,30
RELEASE    vendedor
TERMINATE
*Fin segmento compradores de periódicos
*Inicio segmento compradores de alimentos
GENERATE   300,120
QUEUE      cola_ven
SEIZE      vendedor
DEPART     cola_ven
ADVANCE    240,150
RELEASE    vendedor
TERMINATE
*Fin segmento compradores de alimentos
GENERATE   50400
TERMINATE 1
START      1
```

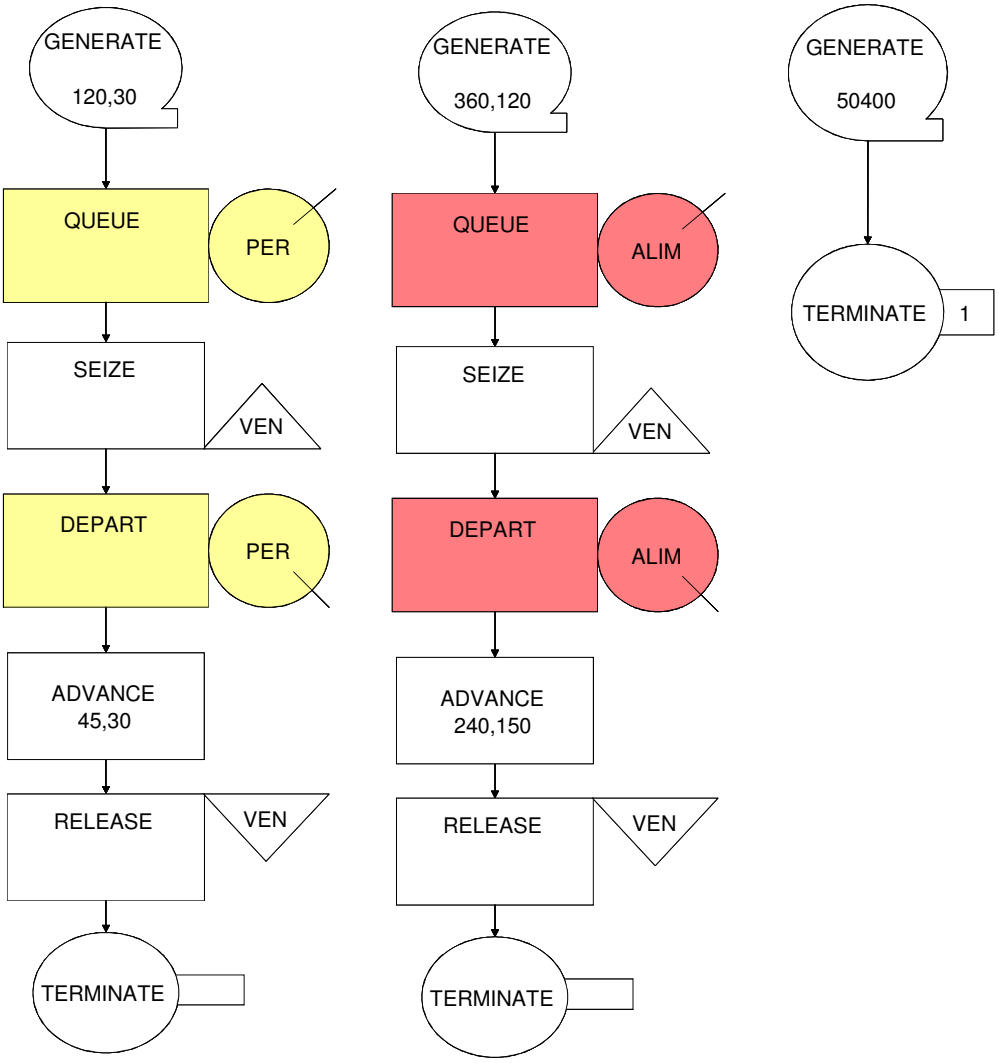
Tiempo medio de espera en cola de un cliente descontando los que no esperan

Tiempo medio de espera en cola de un cliente

Número medio de personas en cola

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
COLA_VEN	76	76	584	3	40.363	3483.404	3501.391	0

Ejemplo con diferentes usuarios, mismo recurso y distintas colas (Continuación)



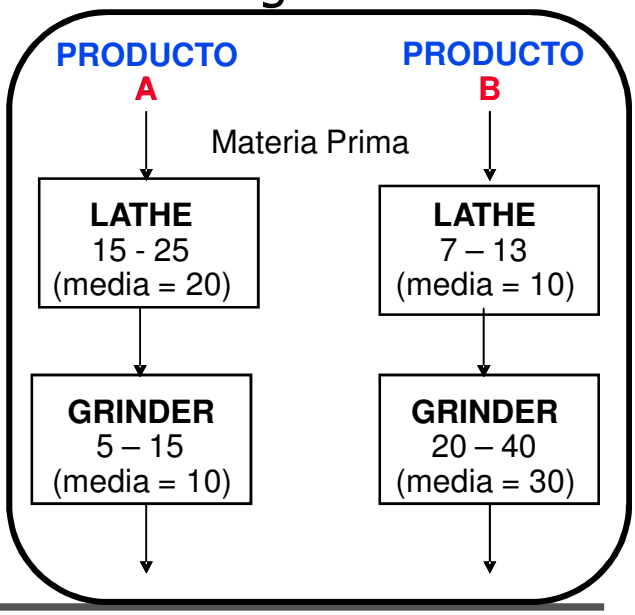
Resultados:

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
VENDEDOR	508	0.994	98.591	1	508	0	0	0	76

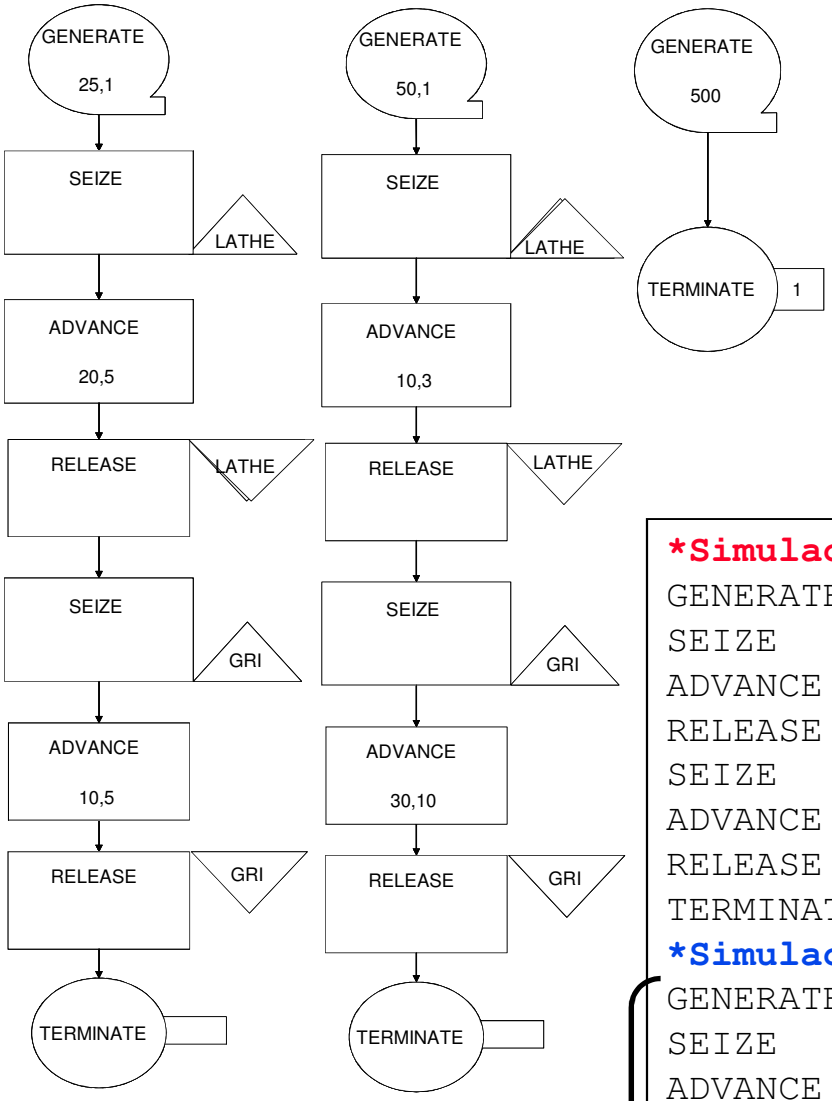
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
COLA_PER	54	54	418	2	29.122	3511.370	3528.251	0
COLA_ALIM	22	22	166	1	11.241	3412.986	3433.671	0

Ejemplo con un tipo único de cliente utilizando sucesivamente varios recursos

- Un taller elabora dos productos A y B
- Cada producto pasa primero por el torno (LATHE) y después por la pulidora (GRINDER)
- Los productos llegan al inicio del proceso con una distribución uniforme [24, 26] para el producto A y con una distribución uniforme [50,51] para el producto B
- Cada producto A requiere de un tiempo de procesado por la LATHE distribuido según una uniforme [15, 25] y de un tiempo de procesado por la GRINDER distribuido según una uniforme [5, 15]
- Cada producto B requiere de un tiempo de procesado por la LATHE distribuido según una uniforme [7, 13] y de un tiempo de procesado por la GRINDER distribuido según una uniforme [20, 40]



Ejemplo con un tipo único de cliente utilizando sucesivamente varios recursos




```
*Simulación Producto A
GENERATE   25,1
SEIZE      LATHE
ADVANCE    20,5
RELEASE    LATHE
SEIZE      GRINDER
ADVANCE    10,5
RELEASE    GRINDER
TERMINATE
*Simulación Producto B
GENERATE   50,1
SEIZE      LATHE
ADVANCE    10,3
RELEASE    LATHE
SEIZE      GRINDER
ADVANCE    30,10
RELEASE    GRINDER
TERMINATE

GENERATE   500
TERMINATE  1
START      1
```

No interesan resultados de la cola por eso no aparecen los bloques QUEUE y DEPART

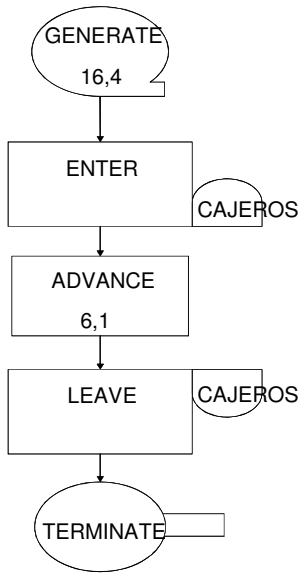
RECURSO MÚLTIPLE
STORAGE ENTER A,B – LEAVE A,B

- Permite modelar un centro de servicio que dispone de varios servidores de similares características
- Para utilizar el bloque STORAGE es necesario utilizar tres tipos de instrucciones:
  - Definición del almacén (antes de GENERATE)

    ej: A STORAGE 2 (2 servidores)
  - Bloque ENTER A, B (ocupa B servidores del recurso A)
  - Bloque LEAVE A, B (libera B servidores del recurso A)
- Los bloques ENTER-LEAVE son análogos a SEIZE-RELEASE para un servidor unitario (forma una cola FIFO única con *s* servidores)
- Ejemplo: La sucursal bancaria dispone de dos empleados en la CAJA

```
CAJEROS   STORAGE   2
          GENERATE  16,4
          ENTER     CAJEROS
          ADVANCE   6,1
          LEAVE     CAJEROS
          TERMINATE

          GENERATE  540
          TERMINATE 1
          START     1
```

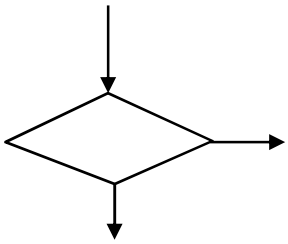
Ponerlo siempre al inicio del segmento



% de utilización de 1 servidor

Número medio de servidores ocupados

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE. C.	UTIL.	RETRY	DELAY
CAJEROS	2	2	0	1	33	1	0.364	0.182	0	0



TRANSFER A,B,C,D

- TRANSFER se utiliza para que las transacciones que pasen por ella puedan saltar a otro bloque
- Tiene 4 operandos que definen tipos de salto:
  - A: Modo de salto ∈{,num,BOTH,ALL,PICK}
  - B: Destino 1
  - C: Destino 2
  - D: Salto
- SALTO INCONDICIONAL: TRANSFER ,B
- SALTO ESTADÍSTICO: TRANSFER A,B,C
  - El valor de A debe estar entre 0 y 1
  - Si el valor muestreado uniforme es superior a A entonces la transacción va a B
  - Si el valor muestreado es inferior va a C
- SALTO SI DESOCUPADO: TRANSFER BOTH,B,C
  - Si la dirección B está disponible la transacción se va a dicha dirección
  - Si en la dirección B tiene un bloque que no está disponible va a la dirección C y si tampoco está disponible espera en el bloque TRANSFER

TRANSFER A,B,C,D

- SALTO CON MÚLTIPLE ELECCIÓN:
  TRANSFER ALL,B,C,D
  - Si el bloque de la dirección B está desocupado salta a él
  - Si está ocupado va comprobando los bloques siguientes hasta la dirección C, el primero que esté disponible salta a él
  - El operando D indica el número máximo de bloques que va saltando en la comprobación. Si se omite el valor por omisión de D es 1.
  - Si no hay ninguno disponible espera en el bloque TRANSFER hasta que alguno lo esté
- SALTO ALEATORIO: TRANSFER PICK,B,C
  - La transacción escoge aleatoriamente una dirección de destino entre B y C
  - En caso de que el bloque de destino estuviera ocupado se queda en TRANSFER esperando a que se desocupe

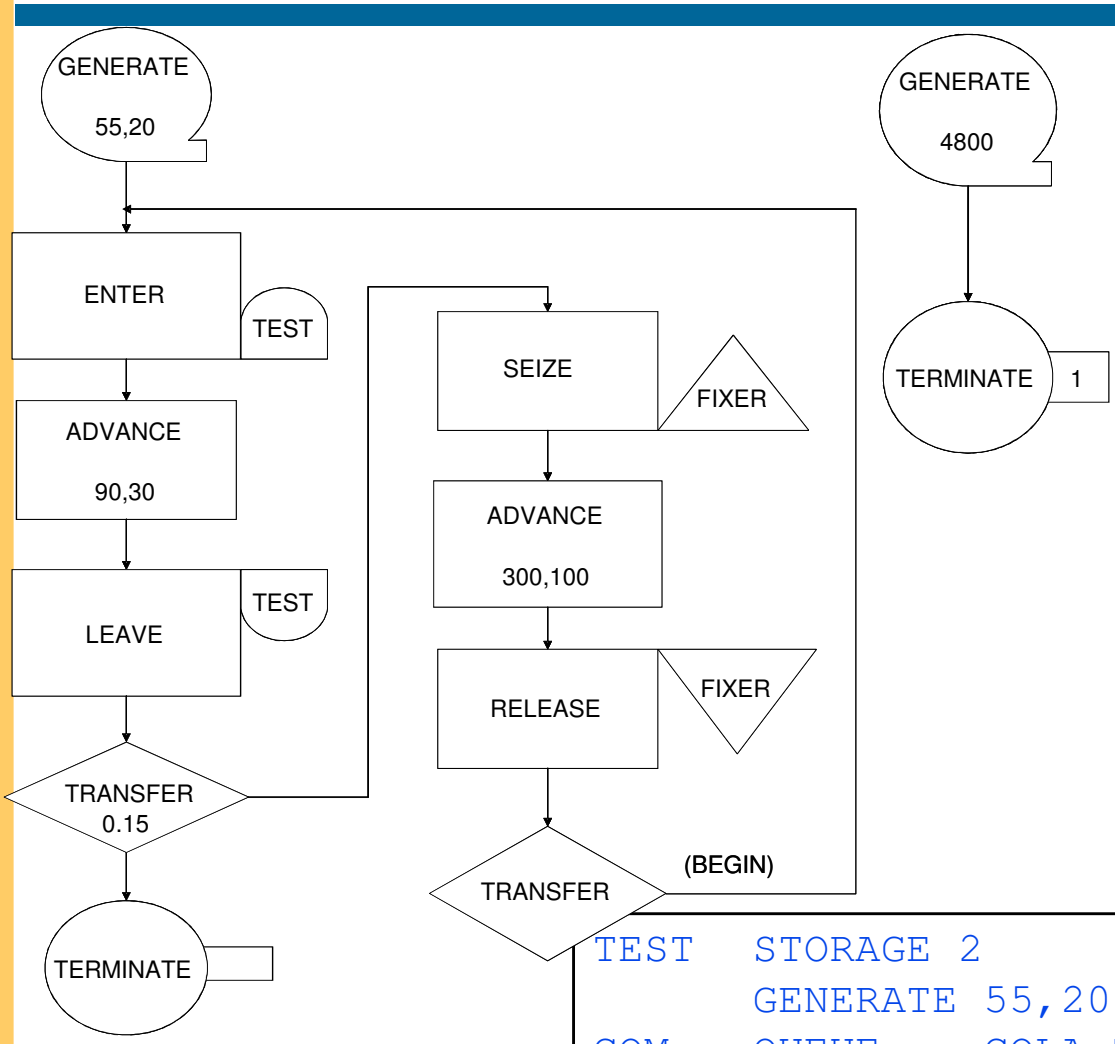
(Este bloque es análogo al GOTO)

TRANSFER (Ejemplo)

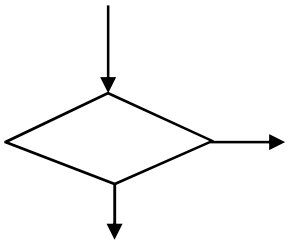
Ejemplo:

- La unidad de control de aparatos de televisión recibe televisores con un tiempo entre llegadas regido por una uniforme [35,75] min
- Los televisores se prueban U[60,120] min
- Se dispone de dos inspectores para probarlos
- El 15% de los aparatos son defectuosos
- Se dispone de un especialista en reparación
- La reparación dura U[200,400] min
- Tras la reparación vuelven a probarse
- La simulación dura 4800 min

TRANSFER (Ejemplo)




```
TEST   STORAGE 2
       GENERATE 55,20
COM    QUEUE    COLA_TEST
       ENTER    TEST
       DEPART   COLA_TEST
       ADVANCE  90,30
       LEAVE    TEST
       TRANSFER 0.15,,FIX
       TERMINATE
FIX    QUEUE    COLA_FIXER
       SEIZE    FIXER
       DEPART   COLA_FIXER
       ADVANCE  300,100
       RELEASE  FIXER
       TRANSFER ,COM
```



GATE O A,B

- GATE altera el flujo normal de una transacción basándose en el estado de un recurso
- Tiene 3 operandos:
  - O: Condición que no cumple el recurso
  - A: Nombre del recurso
  - B: bloque destino cuando el testeo no es exitoso
- Ejemplo: Si un puesto de perritos tiene cola está en uso me voy (siempre hay uno libre)

```
          GENERATE  10,5
          GATE      NU PUESTO_PERRITOS,OTRO
          SEIZE     PUESTO_PERRITOS
          ADVANCE   15
          RELEASE   PUESTO_PERRITOS
OTRO      TERMINATE
```

VALORES DEL ARGUMENTO O:
FNV: La Facility expresada en A debe de estar no disponible
FV: La Facility expresada en A debe de estar disponible
I: La Facility expresada en A debe de estar interrumpida actualmente
NI: La Facility expresada en A debe de estar interrumpida actualmente
NU: La Facility expresada en A no debe de estar en uso
SE: El Storage expresado en A debe de estar vacío
SF: El Storage expresado en A debe de estar lleno
SNE: El Storage expresado en A no debe de estar vacío
SNF: El Storage expresado en A no debe de estar lleno
SNV: El Storage expresado en A debe de estar no disponible
SV: El Storage expresado en A debe de estar disponible
LS: El Logicswitch (llave lógica) debe de estar en estado "on"
LR: El Logicswitch (llave lógica) debe de estar en estado "off"
U: La Facility expresada en operando A debe estar en uso

CREACIÓN GENERALIZADA
GENERATE A,B,C,D,E

- Los operandos A y B son el centro y la mitad de anchura del intervalo de variación (idem a generate simple)
- El operando C define el instante en el que se produce la primera transacción
- El operando D define el número máximo de transacciones que se pueden generar durante el tiempo de simulación
- El operando E define el nivel de prioridad de la transacciones generadas (ver ej. traspare. 29)

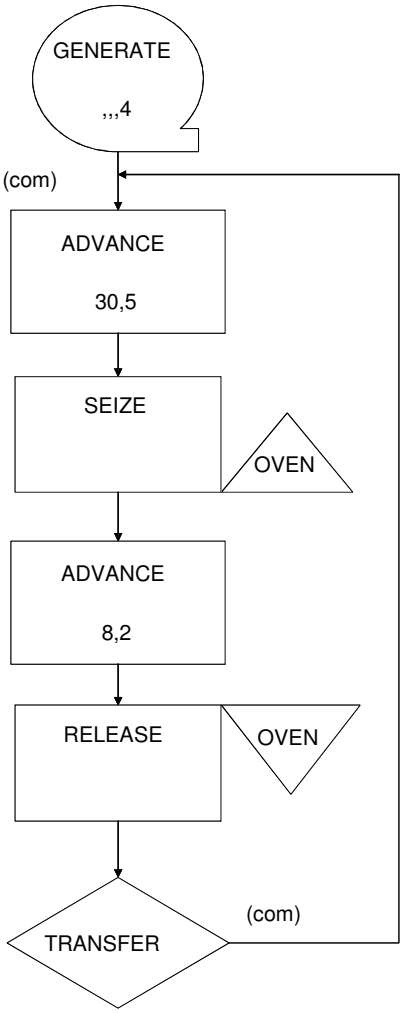
Ejemplo 1: GENERATE ,,,5

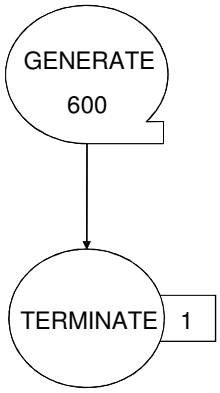
(se generan 5 transacciones al comienzo de la simulación y ninguna más)

Ejemplo 2:

- Una alfarería trabaja 10 horas por día
- Tiene 4 operarios
- Elaborar un jarrón lleva un tiempo aleatorio uniforme en el intervalo [25,35] minutos
- Una vez elaborado ha de adquirir resistencia y se introduce en un único horno de capacidad unitaria durante un tiempo aleatorio uniforme en el intervalo [6,10]
- Una vez fuera del horno el operario puede empezar a elaborar otro jarrón

GENERATE GENERALIZADO (EJEMPLO)



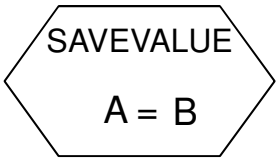



```
com     GENERATE  ,,,4
        ADVANCE   30,5
        SEIZE     OVEN
        ADVANCE   8,2
        RELEASE   OVEN
        TRANSFER  ,com

        GENERATE  600
        TERMINATE 1
        START     1
```

Se generan 4 transacciones en el tiempo=0

ATRIBUTOS NUMÉRICOS (SNA)
W$* Q$* S$*

- Son variables que describen el estado del sistema a lo largo de la simulación
- Atributos de uso habitual (utilizados como args de otras funciones, ver siguiente ejemplo):
  - F$REC1: Indica si el recurso simple REC1 está ocupado (1) o no (0)
  - Q$REC2: Número de transacciones que están en la cola formada frente al recurso REC2
  - W$DIR: indica el nº de personas que están en la dirección DIR
  - S$NOM: Indica el número de transacciones que utilizan el recurso múltiple NOM
  - R$NOM: Capacidad disponible del STORAGE NOM
  - C1: Tiempo de simulación
  - RN#: Obtiene un número aleatorio entre 0 y 999 del generador # de números aleatorios
- Existen muchos más que se describen en el manual



GUARDAVALORES

- Son variables que almacenan valores del modelo (variables globales)
- Se referencian anteponiendo a su nombre X$
- Se asigna su valor utilizando el bloque SAVEVALUE

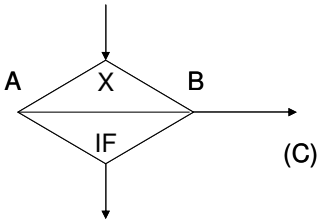
Ejemplo: SAVEVALUE colarec2, Q$REC2

- Si se utiliza el guardavalor como contador se puede usar una expresión compactada

SAVEVALUE ordenadores+,1

ordenadores = ordenadores+1

equivalente a



CONDICIÓN
TEST O A,B,C

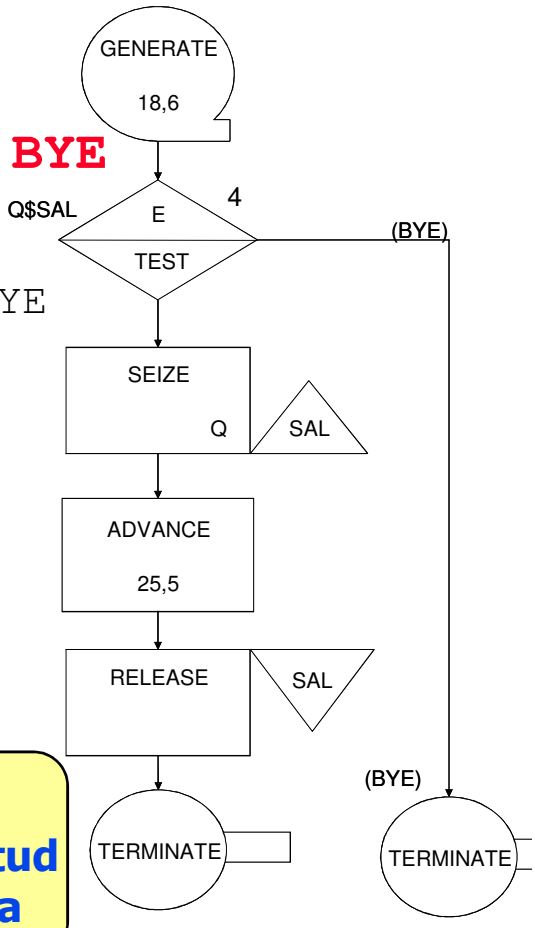
- Este bloque en función del cumplimiento de una condición (análogo a IF-THEN-ELSE) permite:
  - la transacción avance por el segmento si cumple
  - la transacción espere en el bloque si no cumple y no se indica dirección **C**
  - la transacción va a una dirección **C** si **NO** se cumple la condición O entre A y B. Dicha condició O puede ser:

G (>) , L (<), GE (>=), LE (<=), E (=), NE (<>)

- El bloque TEST puede tener como argumentos atributos numéricos

TEST L Q$COLA_SAL,4,BYE

```
        GENERATE 18,6
        TEST L Q$COLA_SAL,4,BYE
        QUEUE COLA_SAL
        SEIZE SAL
        DEPART COLA_SAL
        ADVANCE 25,5
        RELEASE SAL
BYE     TERMINATE

        GENERATE 480
        TERMINATE 1
```



Si la transacción comprueba que la cola del recurso SAL alcanza una longitud de 4 o más entonces se va del sistema

OPERADORES MATEMÁTICOS Y LÓGICOS

OPERADORES MATEMÁTICOS

^ : Exponenciación
# : Multiplicación
/ : División
\ : División entera
@ : Resto entero
- : Resta
+ : Suma

OPERADORES LÓGICOS

>= : Operador *mayor o igual que* : GE
(A >= B ó A GE B devuelve 1 si A es mayor o igual que B y 0 en otro caso).
<= : Operador *menor o igual que* : LE
>: Operador *mayor que* : G
< : Operador *menor que* : L
= : Operador *igual* : E
!= : Operador *no igual* : NE
& : Operador *AND* : AND
| : Operador *OR* : OR

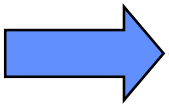
Ejemplo:

```
TEST L ((Q$CAJA = 0) & (C1 > 3600)),1,BYE
```

Continuan debajo del TEST aquellas transacciones que se encuentran sin cola en el recurso CAJA cuando el tiempo de simulación es superior a 3600

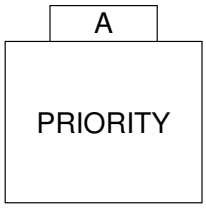
CONTENIDOS

- **Parte general**
  - Segmentos y bloques
  - Creación de transacciones
  - Finalización la simulación
  - Comentarios en el modelo
  - Tiempo de utilización
  - Almacenamiento de resultados
  - Recursos unitarios
  - Colas de espera
  - Tablas de tiempos de espera
  - Recursos de capacidad múltiple
  - Direccionamiento de transacciones
  - Creación generalizada
  - Atributos numéricos
  - Direccionamiento condicional
  - Operadores matemáticos y lógicos



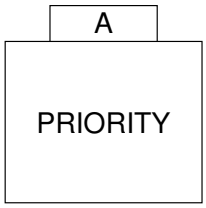
- **Opciones adicionales**
  - Prioridades
  - Interrupciones
  - Funciones aleatorias
  - Variables
  - Funciones adicionales
  - Guardar valores
  - Histogramas
  - Parámetros
  - Seleccionar servidor
  - Loop
  - Separación y unión
  - Obtención de varias muestras

GPSS OPCIONES ADICIONALES
PRIORIDADES

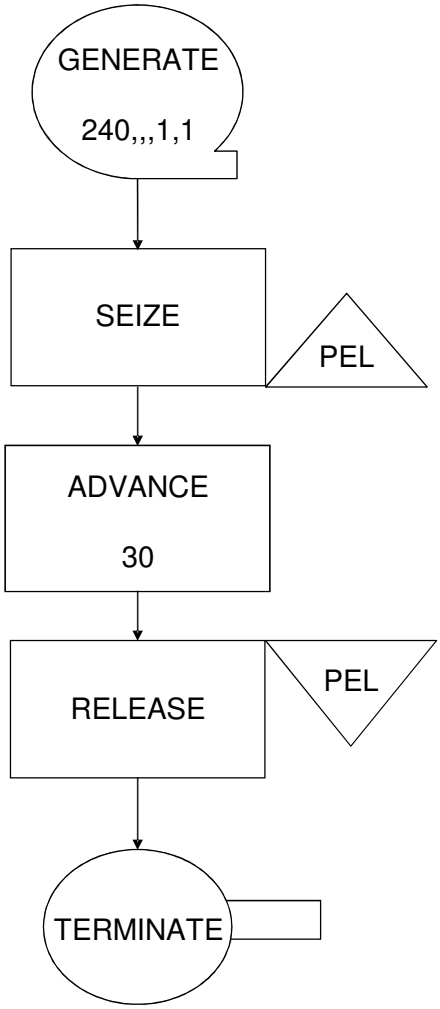


- Sirve para transacciones que deben ser atendidas con antelación a otras, aunque su llegada al recurso sea posterior
- Se consigue generar transacciones con prioridad con el operando E del bloque GENERATE
- Cuanto mayor sea el valor de E mayor será la prioridad de la transacción (< 100.000)
- Se puede cambiar la prioridad de una transacción con el bloque PRIORITY
- El uso de prioridades no da lugar a la interrupción del servicio de transacciones atendidas

PRIORIDAD (Ejemplo)



- Una peluquería con un solo peluquero abre a las 9
- Hace una parada de 30 min para comer tras el primer cliente atendido tras las 13:00
- Los clientes esperan a que vuelva para ser atendidos
- Los clientes llegan según una U[12,24] min y son atendidos según una U[20,30] min




```
*Segmento de Clientes
   GENERATE 18,6
   QUEUE COLA_PELUQUERO
   SEIZE PEL
   DEPART COLA_PELUQUERO
   ADVANCE 25,5
   RELEASE PEL
   TERMINATE
*Segmento de Parada de comida
   GENERATE 240,,,1,1
   SEIZE PEL
   ADVANCE 30
   RELEASE PEL
   TERMINATE
*Segmento de Parada diaria
   GENERATE 480
   TERMINATE 1
   START 1
```

1 transacción con prioridad 1

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
PELUQUERO	18	0.954	25.438	1	19	0	0	0	10

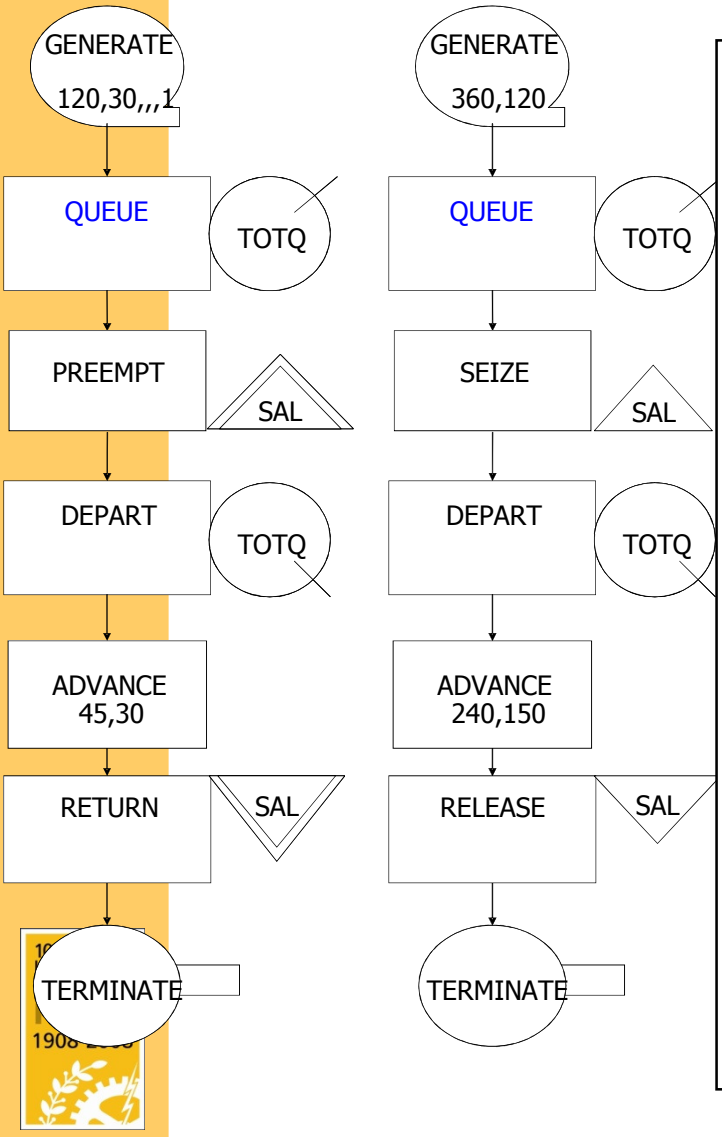
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
COLA_PELUQUERO	11	10	27	1	5.064	90.029	93.492	0

INTERRUPCIONES
PREEMPT A,B – RELEASE A

- La interrupción del servicio puede ser realizada por el uso de la pareja de bloques PREEMPT – RELEASE
- PREEMPT se comporta como un bloque SEIZE cuando el recurso está desocupado
- El operando A es el nombre del recurso SEIZE-RELEASE que puede ser interrumpido (no se puede utilizar con STORAGE)
- En caso de que se omita B si la transacción que usa PREEMPT A se encuentra el recurso ocupado puede:
    - Esperar si el recurso está ocupado por una transacción que ha utilizado PREEMPT
    - Interrumpe el servicio de la transacción si ésta lo ocupó con el bloque SEIZE (clientes menos prioritarios)
- El operando B toma el valor PR si la interrupción se realiza si la transacción que llega tiene mayor prioridad que la transacción que está usando el recurso en ese momento (independientemente de si la transacción ocupó un bloque SEIZE o PREEMPT)
- La transacción interrumpida completa el servicio cuando hayan terminado transacciones con mayor prioridad que estuvieran en cola

INTERRUPCIONES
EJEMPLO 1

- Una tienda vende periódicos y alimentos
- Los clientes de periódicos interrumpen el servicio que el tendero presta a clientes que compran alimentos (en el ejemplo se analizan a la vez las colas individuales y la conjunta con 2 tipos de clientes)




```
*Compradores de periódicos
GENERATE      120,30
QUEUE         TOTQ
PREEMPT       SAL
DEPART        TOTQ
ADVANCE       45,30
RELEASE       SAL
TERMINATE
*Compradores de comida
GENERATE      360,120
QUEUE         TOTQ
SEIZE         SAL
DEPART        TOTQ
ADVANCE       240,150
RELEASE       SAL
TERMINATE
* Segmento de parada
GENERATE      50400
TERMINATE     1
START          1
```

INTERRUPCIONES
EJEMPLO 2

- Además, los familiares del dependiente tienen prioridad sobre los compradores de periódicos y de comida

```
*Compradores de periódicos
generate     120,30,,,1
queue        newsq
queue        totq
preempt              sal,pr
depart       newsq
depart       totq
advance              45,30
release              sal
terminate
*Compradores de comida
generate     360,120
queue        foodq
queue        totq
seize        sal
depart       foodq
depart       totq
advance              240,150
release              sal
terminate
* Familiares con prioridad
generate     1500,200,,,2
queue        vip
queue        totq
preempt              sal,pr
depart       vip
depart       totq
advance              280,100
release              sal
terminate
* Segmento de parada
generate     50400
terminate    1
```

FUNCIONES DEFINIDAS POR EL USUARIO DISCRETAS

- Las funciones definidas por el usuario utilizan el bloque FUNCTION
- Este tipo de bloque se define al principio del código antes del primer GENERATE
- Distribuciones DISCRETAS:

Nombre FUNCTION A,B

Ejemplo:

JUAN FUNCTION RN3,D5

0.1,1/0.25,2/0.45,3/0.75,4/1,5

La función denominada JUAN genera valores discretos 1,2,3,4 y 5 con las probabilidades 0.1, 0.15, 0.20, 0.30 y 0.25 respectivamente

RN3 indica que se utiliza el tercer generador de números aleatorios [0,1]

D5 indica que la función es discreta tomando 5 valores distintos

- Se usa para generar transacciones con un tiempo entre ellas procedente de la función JUAN

GENERATE (FN$JUAN)

- Se usa también para dar valores a variables o parámetros (Ej. SAVEVALUE valor, FN$JUAN)

FUNCIONES DEFINIDAS POR EL USUARIO
CONTINUAS

- Distribuciones CONTINUAS

  Ejemplo:

  TOTO FUNCTION RN2,C5

  0,2/0.15,3/0.4,4/0.80,5/1,8

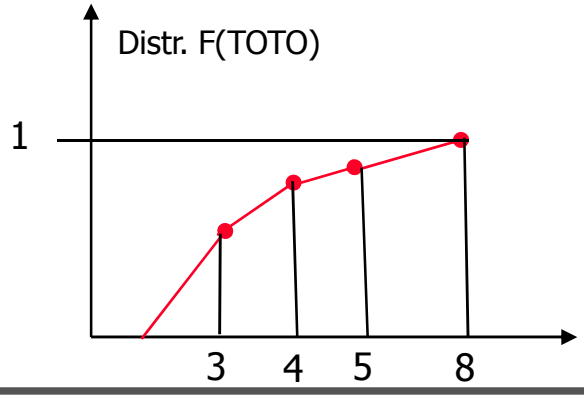
La función TOTO genera valores continuos desde el valor 2 a 8 con una distribución de probabilidad indicada en la tabla adjunta

2 indica que se utiliza el segundo generador de los ocho que tiene GPSS

C5 indica que es un variable continua definida por cinco extremos de los intervalos

- Su utilidad es análoga al de las funciones discretas

Intervalo	$2 \leq TOTO < 3$	$3 \leq TOTO < 4$	$4 \leq TOTO < 5$	$5 \leq TOTO \leq 8$
Prob total	**0.15**	**0.25**	**0.40**	**0.20**



FUNCIONES ALEATORIAS ESTÁNDARES

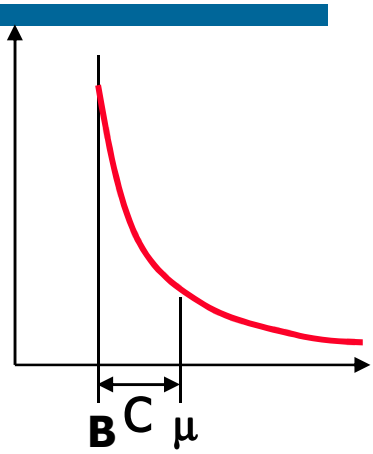
- Exponencial de media C

EXPONENTIAL(A,B,C)

A: Generador de nº aleatorios

B: Parámetro de localización

C: Media de la distribución-B



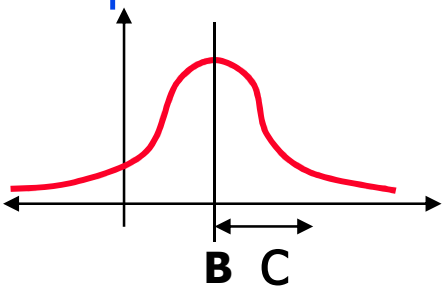
$$f(x) = \frac{1}{C} e^{-\frac{(x-B)}{C}} ; C \geq 0$$

Ej. SAVEVALUE valor, (EXPONENTIAL(1,2,3))

- Normal de media B y desviación típica C

NORMAL (A,B,C)

A: Generador de nº aleatorios

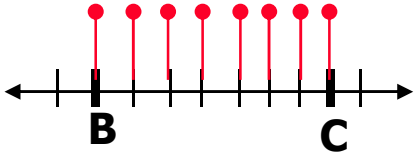


$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-B)^2}{2\sigma^2}}$$

- Uniforme discreta entre B y C

DUNIFORM(A,B,C)

A: Generador de nº aleatorios



$$f(x) = \frac{1}{C-B+1}$$

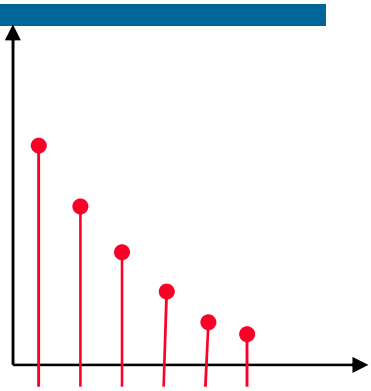
FUNCIONES ALEATORIAS ESTÁNDARES

- Poisson de media B

  POISSON(A,B)

A: Generador de nº aleatorios

$$f(x) = \frac{e^{-B}\, B^{x}}{x!}$$



- Triangular

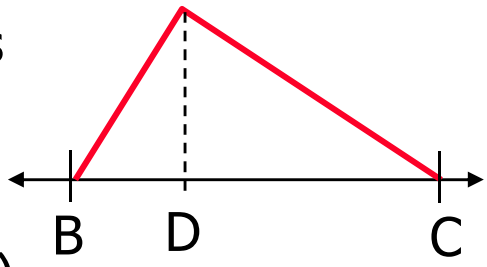
  TRIANGULAR(A,B,C,D)

A: Generador de nº aleatorios

B: Valor mínimo

C: Valor máximo

D: Valor más probable (moda)



$$f(x) = \frac{2(x-B)}{(C-B)(D-B)};\ B \le x \le D$$

$$f(x) = \frac{2(C-x)}{(C-B)(C-D)};\ D \le x \le C$$

- Weibull

  WEIBULL(A,B,C,D)

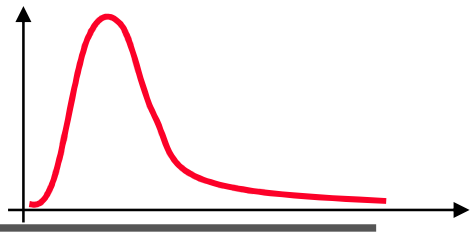
$$f(x) = \frac{D(x-B)^{(D-1)}}{C^{D}} e^{-\left(\frac{x-B}{C}\right)^{D}};\ x > B$$

A: Generador de nº aleatorios

B: Parámetro de localización

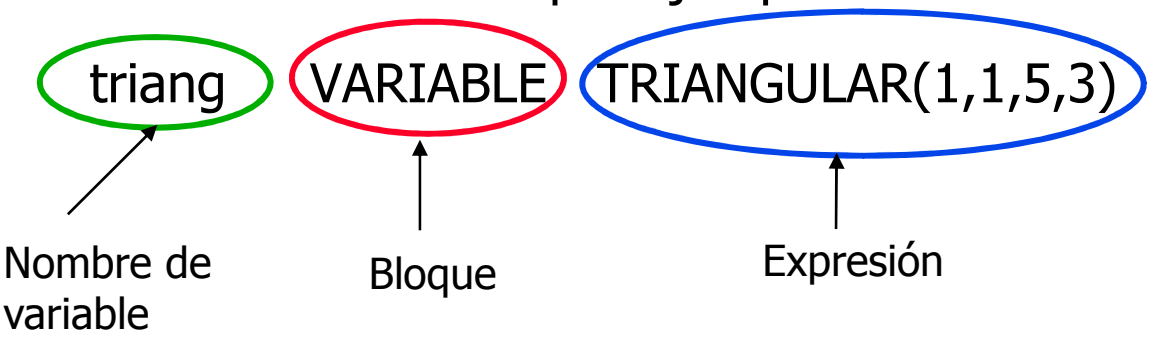
C: Parámetro de escalación

D: Parámetro de forma



VARIABLES y EXPRESIONES

- Las variables evalúan expresiones matemáticas
- Se utilizan cuando la expresión matemática se utiliza frecuentemente en el código
- Se definen antes del primer GENERATE
- Su sintaxis consiste por ejemplo



- Se evalúa su valor incluyendo V$nombre_variable

```
GENERATE        18
SEIZE   SAL
ADVANCE (TRIANGULAR(1,1,5,3))
RELEASE SAL
TERMINATE
GENERATE      480
TERMINATE       1
START   1
```

```
triang VARIABLE (TRIANGULAR(1,1,5,3))
       GENERATE        18
       SEIZE   SAL
       ADVANCE v$triang
       RELEASE SAL
       TERMINATE
       GENERATE 480
       TERMINATE 1
       START 1
```

FUNCIONES ADICIONALES

- Se pueden añadir ATRIBUTOS a las funciones:
  - Q$REC: tamaño de la cola frente a REC
  - S$ALM: contenido actual del STORAGE ALM
  - R$ALM: capacidad libre del STORAGE ALM
  - PR: prioridad de la transacción
  - N$DIR: número total de transacciones que han pasado por la dirección simbólica DIR
  - W$DIR: número actual de transacciones que han pasado por la dirección simbólica DIR
  - C1: valor del reloj relativo de simulación

Ejemplo 1: Dependiendo de los coches que están en la carretera el tiempo del trayecto es mayor o menor

```
TIEMP FUNCTION Q$CTRA, C2
0,300/100,600
```

Ejemplo 2: Se quiere que durante las primeras 4 horas el tiempo medio entre llegadas valga 25 min, durante las 3 horas siguientes valga 15 min y durante las 2 horas siguientes valga 20 min

La función TMEL obtiene los tiempos entre llegadas

```
TMEL FUNCTION C1,D3
240,25/420,15/540,20
```

FUNCIONES ADICIONALES

- ABS(A): valor absoluto
- ATN(A): arco tangente (en radianes)
- DEC(A): parte decimal
- EXP(A): función exponencial
- INT(A): parte entera
- LOG(A): logaritmo neperiano
- SGN(A): signo (0-positivo, 1-negativo)
- SIN(A): seno
- COS(A): coseno
- SQR(A): raíz cuadrada
- TAN(A): tangente (en radianes)

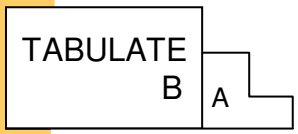
FUNCIONES ADICIONALES
DIRECCIONES

- Existen funciones que obtienen como resultado una dirección del código a partir del uso de números como variables independientes

Ejemplo:

```
ELECC FUNCTION RN5,D3
  0.5,PAN/0.7,CARNE/1,LECHE
      GENERATE 10,2
      TRANSFER ,FN$ELECC
PAN   ADVANCE 6,3
      TERMINATE
CARNE ADVANCE 4,2
      TERMINATE
LECHE ADVANCE 2,1
      TERMINATE

      GENERATE 480
      TERMINATE 1
      START 1
```

Las transacciones se dirigen con una probabilidad del 50% van a la dirección PAN, con un 20% a la dirección CARNE y con un 30% a la dirección LECHE



TABLAS ESTADÍSTICAS
TABLE A,B,C,D
TABULATE A

dirección TABLE A,B,C,D

- Se escribe antes del primer GENERATE
- La dirección indica el nombre de la tabla
- El operando A es un atributo numérico sobre el cual queremos obtener un histograma
- El operando B es el límite superior del intervalo inferior
- El operando C es la amplitud de cada intervalo
- El operando D es el número de intervalos

TABULATE dirección

- El bloque TABULATE tiene un operando A que es el nombre de la tabla
- Cada vez que una transacción pasa por el bloque TABULATE se incorpora el valor del atributo numérico a la tabla
- Generaliza de alguna manera el bloque QTABLE

TABLAS ESTADÍSTICAS
TABLE A,B,C,D
TABULATE A,B (Ejemplo)

Ejemplo:

- Una tienda abre 8 horas al día
- La llegada de clientes se produce uniformemente entre [8,12] minutos
- Existe un único dependiente que dedica a cada cliente un tiempo entre [7,13]
- Se quiere tabular el tiempo de espera en ser atendido con intervalos de 1 minuto

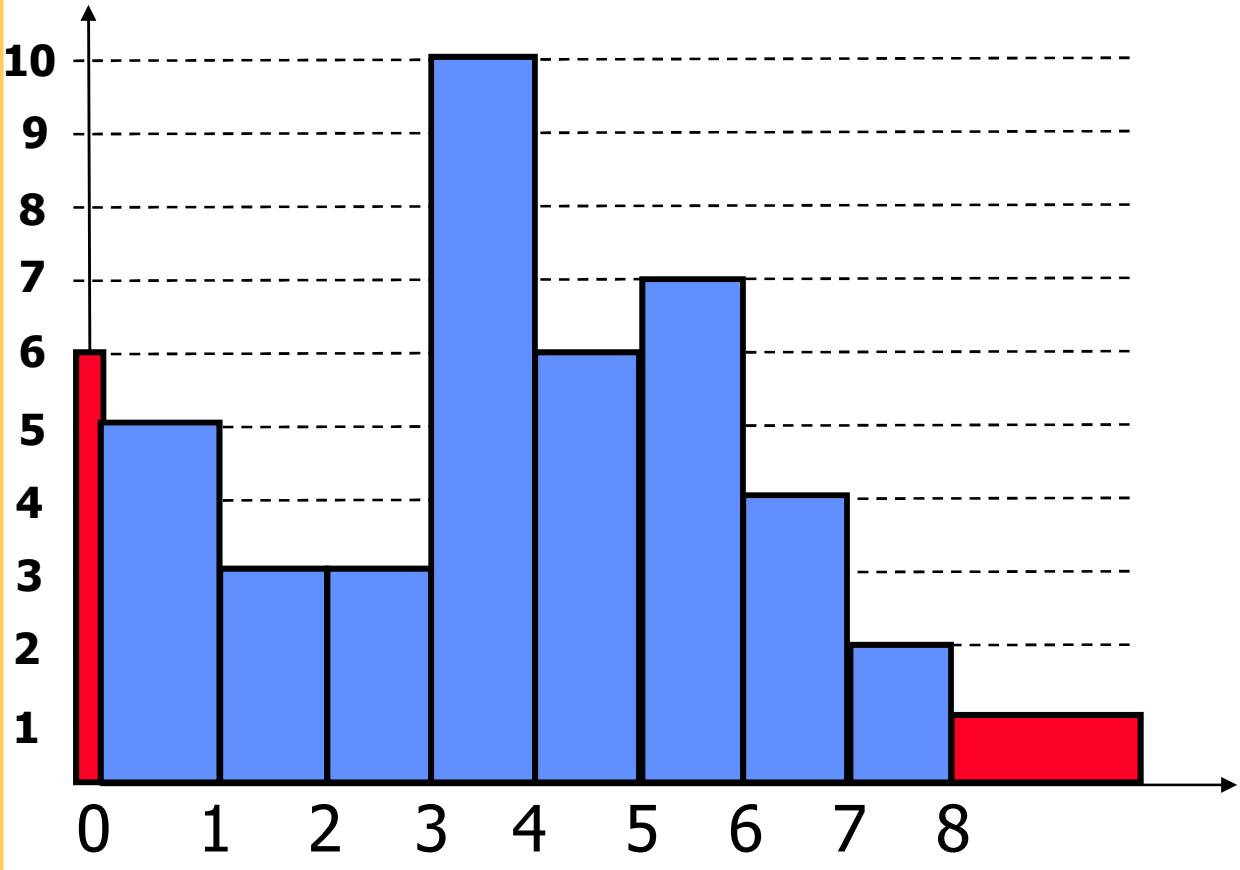
```
tespera   TABLE v$tiempo,0,1,10    ;Histograma
tiempo    VARIABLE x$tiempo2-x$tiempo1 ;Diferencia

* Segmento de atención a clientes
        GENERATE 10,2          ;clientes [8,12]
        SAVEVALUE tiempo1,C1 ; tiempo de entrada
        QUEUE cola_espera      ;Se entra en cola
        SEIZE vendedor       ;El cliente es atendido
        DEPART cola_espera     ;Se sale de la cola
        SAVEVALUE tiempo2,C1 ;Tiempo de salida
        TABULATE tespera       ;Se tabula
        ADVANCE 10,3         ;tiempo servicio [7,13]
        RELEASE vendedor       ;El vendedor se libera
        TERMINATE     ;El cliente se va del sistema
* Segmento de horario de atención de la tienda
        GENERATE 480           ;La tienda abre 8 horas
        START 1                ;Solo se simula 1 vez
        TERMINATE 1
```

¿Porqué no se está calculando correctamente el tiempo de espera?

TABLAS ESTADÍSTICAS
TABLE A,B,C,D
TABULATE A,B (Ejemplo)

Ejemplo (continuación):

RANGE			RETRY	FREQUENCY	CUM.%
_	-	0.000		6	12.77
0.000	-	1.000		5	23.40
1.000	-	2.000		3	29.79
2.000	-	3.000		3	36.17
3.000	-	4.000		10	57.45
4.000	-	5.000		6	70.21
5.000	-	6.000		7	85.11
6.000	-	7.000		4	93.62
7.000	-	8.000		2	97.87
8.000	-	_		1	100.00

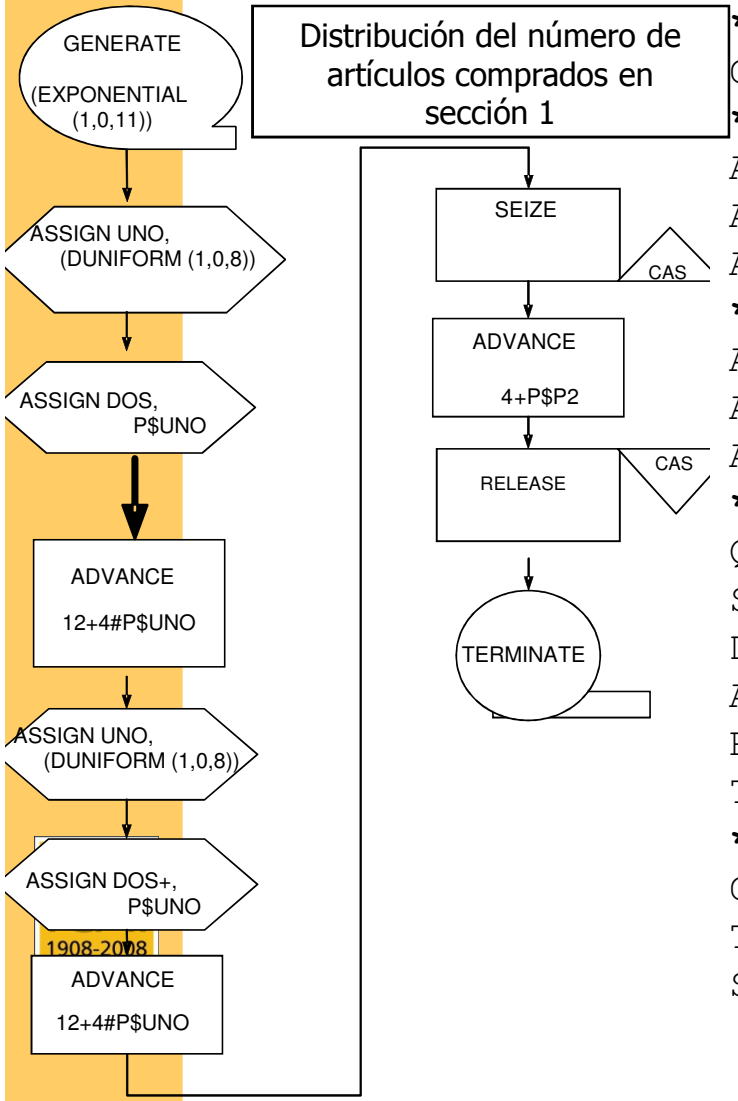


PARÁMETROS
ASSIGN A,B,C

- Son características propias de las transacciones (variable local a nivel de transacción: como la prioridad, que es una característica de cada transacción)
- Pueden ser identificados con caracteres alfanuméricos (habitualmente se numeran)
- El valor asignado a dichos parámetros es propio de cada transacción
- El operando A identifica el nombre del parámetro, el operando B identifica el valor asignado al parámetro
- El operando opcional C se multiplica por B y su valor se asigna al parámetro
- Para hacer referencia al valor de los parámetros es necesario utilizar P$ (en SAVEVALUE era X$)
- Para multiplicar su valor por una constante u otro parámetro se utiliza el símbolo #
- El parámetro PR (prioridad) se refiere a la prioridad de la transacción, pudiéndose modificar su valor a lo largo de la simulación

PARÁMETROS (Ejemplo 1)

- Un supermercado dispone de dos secciones
- Los clientes llegan según una exponencial de media 12 min
- Los clientes pasan un tiempo distinto en cada sección dependiendo del número de artículos comprados en cada sección: 12 minutos más 4 minutos por artículo comprado
- Tras pasar por ambas sección en la caja tardan en cobrar los artículos comprados en ambas secciones un tiempo mínimo de 2 minutos más 0.1 minutos por artículo comprado.




```
*Llegada de clientes
GENERATE (EXPONENTIAL(1,0,12))
*Primera sección
ASSIGN UNO,(DUNIFORM(1,0,8))
ASSIGN DOS,P$UNO
ADVANCE       (12+4#P$UNO)
*Segunda sección
ASSIGN UNO,(DUNIFORM(1,0,9))
ASSIGN DOS+,P$UNO
ADVANCE       (12+4#P$UNO)
*Cajero
QUEUE  COLA_CAJERO
SEIZE  CAJERO
DEPART COLA_CAJERO
ADVANCE       (2+0.1#P$DOS)
RELEASE       CAJERO
TERMINATE
*Segmento de parada
GENERATE      3600
TERMINATE     1
START  1
```

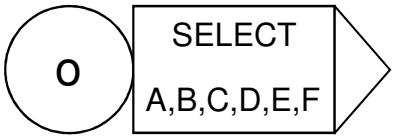
BLOQUE MARK A

- Se utiliza para almacenar el tiempo en que una transacción pasa por ella
- El tiempo queda almacenado en el parámetro A del bloque MARK

Ejemplo: El tiempo que tarda una persona en pagar en una tienda es proporcional al momento del día en que ha llegado e inversamente proporcional al número de clientes que han entrado en la tienda hasta ese momento

```
GENERATE (TRIANGULAR(1,25,50,30))
SAVEVALUE TRANSACCIONES+,1
MARK ENTRADA
SEIZE CAJA
ADVANCE (P$ENTRADA/X$TRANSACCIONES)
RELEASE CAJA
TERMINATE 1
```

SELECCIONAR
SELECT O,A,B,C,D,E,F



- Este bloque es utilizado habitualmente para elegir colas o recursos
- Los operandos O,A,B,C son de uso obligatorio
- O es un operador condicional u operador lógico, por ejemplo: MAX (se va a la cola más grande), MIN, NU (not used),…
- A es el parámetro de la transacción en el cual se almacena la elección
- B y C son el número menor y mayor de la entidades entre las que se va a elegir (por tanto los nombres de las entidades deben ser números)
- D es el valor de referencia del operando E cuando se utiliza el bloque en modo condicional (ver trasparencia anterior)
- E indica la naturaleza de las entidades a seleccionar: S (recursos múltiples-STORAGE), F (recurso unitario-SEIZE-RELEASE), Q (tamaño de cola)
- F indica la dirección a la cual se envía la transacción si no se selecciona entidad alguna

SELECT
Ejemplo

- En un servicio de atención de Telepizza se dispone de 7 telefonistas
- Con cada cliente se tarda un tiempo uniformemente distribuido entre [3,7] minutos
- Si no hay teléfonos libres la llamada queda en espera en el teléfono con menor cola hasta que es atendido
- Los clientes llaman al servicio de atención con un tiempo entre llamadas consecutivas distribuido según una uniforme [1,3] minutos

```
            GENERATE 2,1
*Elección de teléfono o encolamiento(q_tel)
            SELECT NU telefono,1,7,,F,q_tel
            TRANSFER ,atencion
q_tel       SELECT MIN telefono,1,7,,Q;Elección de cola
atencion    QUEUE P$telefono     ;Cola teléfono elegido
            SEIZE P$telefono     ;Servicio en teléfono
            DEPART P$telefono    ;Abandono de cola
            ADVANCE 5,2          ;Demora de llamada
            RELEASE P$telefono   ;Fin llamada
            TERMINATE 1
```

SELECT
OPERADORES LÓGICOS

A continuación se indica la condición para que el recurso sea seleccionado:

Valores de O:

FNV → La facility ha de estar no disponible

FV → La facility tiene que estar disponible

I → La facility tiene que estar interrumpida

NI → La facility no tiene que estar interrumpida

NU → La facility no tiene que estar en uso

SE → El storage tiene que estar vacio

SF → El storage tiene que estar lleno

SNE → El storage no tiene que estar vacio

SNF → El storage no tiene que estar lleno

SV → El storage tiene que estar disponible

SNV → El storage no tiene que estar disponible

SELECT
OPERADORES CONDICIONALES

A continuación se indica la condición para que el recurso sea seleccionado (sólo aplicable a la cantidad de transaciones en la cola)

Valores de O:

E → (Operando E)=(Operando D)

G → E>D

GE → E>=D

L → E<D

LE → E<=D

MAX → Máximo valor del operando E

MIN → Mínimo valor del operando E

NE → El operando E debe ser distinto a D

EJEMPLO:

```
SELECT G, caja,1,3,10,Q,salir
```

De entre las colas 1 a 3 se selecciona la primera que tenga una cola mayor que 10

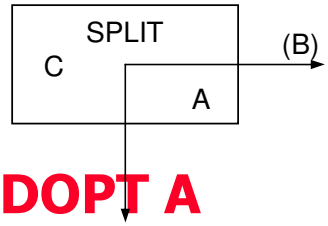
LOOP
LOOP A,B

- LOOP es utilizado para controlar el número de veces que una transacción ha de pasar por un determinado conjunto de bloques
- Tiene 2 operandos:
  - A: cierto parámetro de la transacción que es decrementado en uno en cada iteración
  - B: dirección a la que se salta una vez decremento el parámetro (si el parámetro es 0 no se produce salto)
- Cuidado el parámetro no puede tomar valores negativos pues GPSS daría un error de simulación

LOOP
Ejemplo

- Supermercado

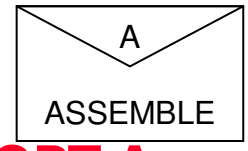
```
*Veamos como una persona hace un ADVANCE por
*las góndolas (mobiliario exhibición) tantas
*veces como artículos compra
            ASSIGN      AUX,P$CANTART
            ADVANCE     180,6
OTRGON:     ADVANCE     45,15
            LOOP        AUX,OTRGON

*La selección de la caja es inteligente
*(la desocupada o cola mínima)
            SELECT NU cola,1,X$CAJAS,,,OCUP
            TRANSFER ,PAGAR
OCUP:       SELECT MIN cola,1,X$CAJAS,,Q

*En caja el cajero al atenderlo también repite
*un advance(tiempo de registro del artículo)
*tantas veces como artículos compró el cliente.
*Luego hay un advance que representa el
*tiempo en pagar
PAGAR:    QUEUE       *cola
          SEIZE       *cola
          DEPART      *cola
          ASSIGN      AUX, P$CANTART
OTRREG:   ADVANCE     8,4
          LOOP AUX,OTRREG
          ADVANCE 60,20
          RELEASE   *cola
```



DIVIDIR Y AGRUPAR

SPLIT A,B,C ASSEMBLE A ADOPT A

- SPLIT realiza tantas copias de transacciones como indique el operando A (copias idénticas)
- El operando B indica la dirección del bloque al cual se dirigen dichas copias
- El operando C indica el número de parámetro con el que se indica el número de copia + 1 de las transacciones (permite distiguir unas copias de otras). Ejemplo: SPLIT 4,CAMINO2,12
- ASSEMBLE agrupa en una transacción un número de transacciones indicado por el operando A
- Estas transacciones han sido previamente creadas con SPLIT
- En el proceso de agrupamiento la transacción agrupada conserva el valor de los parámetros de la última transacción que llegó al bloque ASSEMBLE
- ADOPT indica el grupo de ensamblaje al cual pertenece la transacción. Sólo se agrupan transacciones del mismo grupo



DIVIDIR Y AGRUPAR

SPLIT A,B,C ASSEMBLE A ADOPT A

- Ejemplo:

Un proyecto consiste en la reforma de una casa y se considera que esta constituido por cuatro actividades:

A: Albañilería

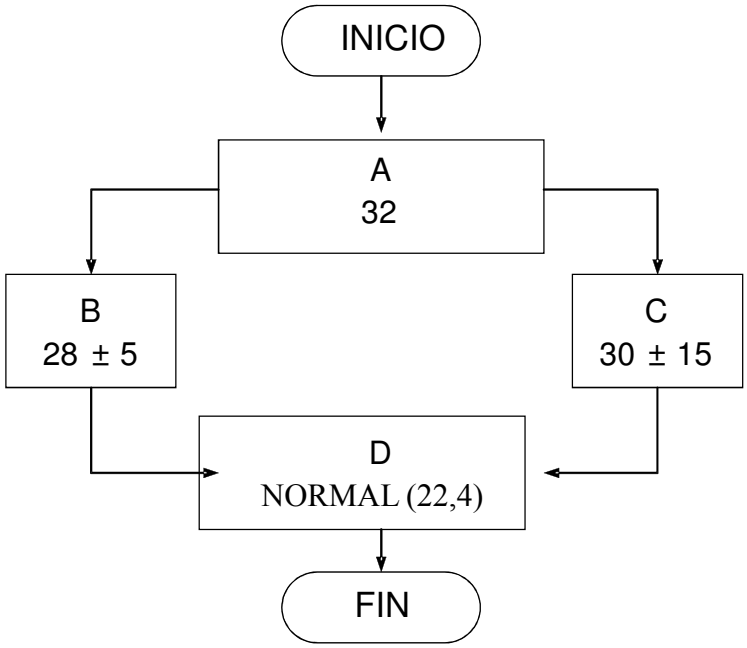
B: Fontanería

C: Instalación eléctrica

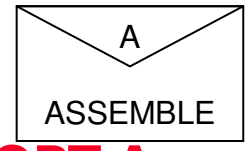
D: Pintura

de forma que A precede a B y C, y éstas, a su vez, preceden a D.

El diagrama de actividades con sus tiempos en horas:



Simular el proceso de construir 100 casas



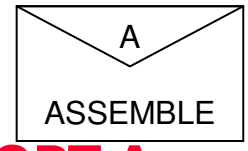
DIVIDIR Y AGRUPAR
SPLIT A,B,C ASSEMBLE A ADOPT A

- Ejemplo (código):

```
GENERATE    ,,,100
      SAVEVALUE CONT+,1
      ADOPT X$CONT
*Actividad A: Estructura
      SEIZE     estructura
      ADVANCE   32
      RELEASE   estructura
      SPLIT     1,elec
*Actividad B: Fontanería
      SEIZE     fontaneria
      ADVANCE   28,5
      RELEASE   fontaneria
      TRANSFER  ,pint
*Actividad C: Instalación eléctrica
elec  SEIZE     electricidad
      ADVANCE   30,15
      RELEASE   electricidad
*Actividad D: Pintura
pint  ASSEMBLE  2
      SEIZE     pintura
      ADVANCE   (NORMAL(1,22,4))
      RELEASE   pintura

      TERMINATE 1
      START     100
```

100 transacciones al inicio

La copia se va a elec, el padre sigue por B



DIVIDIR Y AGRUPAR

SPLIT A,B,C ASSEMBLE A ADOPT A

- Resultado (código):

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	3272.752	17	4	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	SAVEVALUE	100	0	0
	3	ADOPT	100	0	0
	4	SEIZE	100	0	0
	5	ADVANCE	100	0	0
	6	RELEASE	100	0	0
	7	SPLIT	100	0	0
	8	SEIZE	100	0	0
	9	ADVANCE	100	0	0
	10	RELEASE	100	0	0
	11	TRANSFER	100	0	0
ELEC	12	SEIZE	100	0	0
	13	ADVANCE	100	0	0
	14	RELEASE	100	0	0
PINT	15	ASSEMBLE	200	0	0
	16	SEIZE	100	0	0
	17	ADVANCE	100	0	0
	18	RELEASE	100	0	0
	19	TERMINATE	100	0	0

100 años de Ingeniería

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.
ESTRUCTURA	100	0.978	32.000	1
FONTANERIA	100	0.855	27.976	1
ELECTRICIDAD	100	0.875	28.635	1
PINTURA	100	0.674	22.048	1

CAMBIO DE SEMILLA
RMULT A,B,C,D,E,F,G

•Para que un generador RN devuelva distintos números aleatorios en diferentes simulaciones, GPSS dispone de una herramienta que permite alterar la semilla

•Cada argumento va asociado a cada uno de los siete primeros generadores por orden secuencial siendo "A" el RN1 y "G" el RN7

•Por ejemplo si se quieren hacer dos diferentes simulaciones en un modelo que utiliza el RN5, basta con especificar en cada una de ellas un factor diferente en la sentencia RMULT. Por ejemplo:

En la primera simulación añadir → RMULT ,,,,35
En la segunda simulación añadir→ RMULT ,,,,460

De esta manera se habrá variado la secuencia de números aleatorios y se obtendrán distintos resultados en la simulación

•En los bloques GENERATE, ADVANCE y TRANSFER el generador utilizado es el RN1