

# REPÚBLICA ARGENTINA

## UNIVERSIDAD NACIONAL DE GENERAL SARMIENTO

Instituto de Industrias

Cátedra de Investigación Operativa

## UNIVERSIDAD NACIONAL DE LOMAS DE ZAMORA

Instituto de Industrias

CÁTEDRA DE INVESTIGACION OPERATIVA

## UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Concepción del Uruguay

CÁTEDRA DE SISTEMAS DE GESTIÓN II

# SIMULACIÓN

Puede reproducirse libremente. Se agradecerá citar la fuente.

*Claudio L. R. Sturla*

Podríamos imaginar una simulación computerizada de la acción de los péptidos en el hipotálamo, precisa y exacta hasta la última sinapsis. Pero igualmente podríamos imaginar una simulación de la oxidación de los hidrocarburos en el motor de un automóvil o de la acción de los procesos digestivos en un estómago que está digiriendo pizza. Y la simulación no es más real en el caso del cerebro que lo es en el del motor o del estómago. Descartados los milagros, no podríamos hacer funcionar nuestro automóvil efectuando una simulación informática de la combustión de la gasolina, y no se digiere una pizza ejecutando el programa que simula tal digestión.

John R. Searle  
Universidad de California  
Berkeley

📖 ➡ Bibliografía:

- Khoshnevis, Behrok, Discrete Systems Simulation, McGraw Hill, 1.994, ISBN 0-07-833302-4.

## Sistemas

Un sistema es una parte de la realidad.

Por eso cualquier cosa puede ser pensada como un sistema.

Una cosa en común entre todos los sistemas es que los componentes interactúan unos con otros. La naturaleza de esas interacciones está determinada por las leyes físicas que gobiernan su comportamiento.

Todos los sistemas pueden ser considerados componentes de otro sistema de un nivel más alto (con la posible excepción del universo en sí mismo).

Dado que las partes del sistema interactúan unas con otras, esto nos lleva al hecho de que cada sistema es influenciado por otros.

O sea, ningún sistema está totalmente aislado de las influencias externas.

Si las influencias externas tienen un impacto tan grande que afecta en forma muy significativa al sistema en consideración, los límites en consideración debe expandirse para incluir estas influencias.

Por ejemplo, el diseño de una planta nuclear debe incluir el subsistema socio-político o el sistema completo estará seriamente incompleto.

**Si las influencias externas son considerables, pero su pauta de comportamiento puede ser identificada utilizando un método fiable de observación sin necesidad de explorar la estructura interna de los sistemas externos que generan las influencias, entonces las influencias pueden ser tratadas con "inputs" del sistema en estudio.**

Estos "inputs" pueden ser valores fijos, información tabulada o relaciones funcionales.

Si las influencias externas son insignificantes, pueden ser ignoradas, al menos inicialmente.

Qué influencia es significativa y cuál no lo es es una pregunta no trivial y puede ser contestada sólo después de que se establezca el propósito del estudio.

Consideremos el problema de una intersección de calles con el propósito de establecer el mejor timing para el cual el tiempo de espera promedio para todos los autos que llegan a la intersección sea minimizado.

Se ve claramente que el comportamiento de la intersección (el foco del estudio) no es independiente de los timings de los semáforos de las intersecciones adyacentes.

El analista puede elegir tratar el arribo de autos como inputs del sistema.

El o ella pueden entonces utilizar los métodos de observación estadística para caracterizar esas pautas de arribos.

Esta forma de trabajar evita al analista la necesidad de estudiar el comportamiento interno de las intersecciones adyacentes.

Si, dada la significación del impacto del timing de los semáforos de las intersecciones adyacentes, falla el esfuerzo para caracterizar las pautas de arribos de autos, el analista debe expandir los bordes del sistema en estudio para incluir las intersecciones adyacentes.

La ventaja de esta expansión es el mayor realismo en el conocimiento del comportamiento del sistema. Resulta en un análisis más preciso y puede llevar a un estudio que mejore la performance general del sistema (tal como la sincronización de varios semáforos).

La desventaja de expandir los límites del sistema es el aumento de complejidad que algunas veces hace que el sistema sea difícil, y hasta imposible, de estudiar con las herramientas de análisis disponibles.

En muchos casos la expansión de los límites del sistema es innecesaria, y algunos factores simplemente podría ser excluidos.

Por ejemplo, en el caso de la intersección supongamos que una de las calles que llega a la intersección conecta con una autopista a una localidad distante.

Supongamos que la autopista pasa por un aeropuerto.

Obviamente, por causa de estas relaciones, el arribo de autos a la intersección está influenciado hasta cierto punto por los horarios de vuelos al aeropuerto, pero esta influencia es remota y lo suficientemente pequeña como para justificar su exclusión del estudio.

Debe notarse que la que puede ser considerada una influencia insignificante para un observador puede ser muy diferente para otro.

Por lo tanto, los dos observadores pueden elegir diferentes límites del sistema y más: pueden tener diferentes representaciones de lo que puede ser el mismo sistema.

Por ejemplo, uno puede centrarse en una clase con el propósito de estudiar el número de categorías (jóvenes, mayores, etcétera) de estudiantes sentados y los factores que afectan esas variables, pero los otros pueden centrarse en el estudio de los cambios de temperatura de la habitación a lo largo del tiempo.

Los límites del sistema, sus componentes y las relaciones entre éstos pueden ser muy diferentes en vista de estos análisis.

Esto nos lleva a una definición de sistema:

**Un sistema es una sección de la realidad que es el foco primario de estudio y está compuesto de partes que interactúan entre sí de acuerdo con ciertas reglas dentro de un límite identificado por el propósito del estudio. Un sistema puede realizar una función no realizable por los componentes individuales.**

## Clasificación de los sistemas

Las medidas que caracterizan a los sistemas son los parámetros y las variables.

Los **parámetros** son medidas independientes que configuran las condiciones de los inputs y la estructura del sistema.

Los parámetros en sistemas hechos por el hombre son controlables directamente.

Las **variables** son medidas que dependen de los parámetros y de las otras variables.

Por ejemplo, el peso y longitud de un péndulo son parámetros, su velocidad dentro de su rango de oscilación es una variable que depende de su peso, longitud, y otras variables (tales como su posición).

El número de máquinas en un taller es también un parámetro (siempre que no cambie durante el curso del estudio); el número de partes esperando ser procesadas y sus tiempos de espera son variables.

El conjunto de algunas variables prescritas en un sistema en algún punto en el tiempo es llamado el **estado del sistema** en ese punto del tiempo.

**El estado de un problema puede definirse como un conjunto estructurado de informaciones que traducen propiedades y parámetros y permiten describir los objetos y las relaciones entre esos objetos en un momento dado y dentro de los límites útiles para la resolución de ese problema.**

Tomándolo de una definición de la física:

"El estado de un cuerpo está definido por el conjunto de propiedades específicas, tales como la densidad, la resistencia eléctrica, el índice de refracción, etc., bastando que se modifique una cualquiera para que pueda decirse que el cuerpo se transforma o que pasa de un estado a otro."

"Los cambios de estado pueden producirse de un modo continuo, variando gradualmente alguna propiedad, como sucede con la dilatación térmica, o de un modo discontinuo, como ocurre con la fusión, por ejemplo. El conjunto de los estados que pueden ser adoptados por un cuerpo, pasando de uno a otro de un modo continuo, constituye un *estado de agregación*. Cuando cambia bruscamente alguna de las propiedades del cuerpo, se dice que experimenta un cambio de estado de agregación. Los estados de agregación más importantes son el gaseoso, el líquido y el sólido cristalino."

"Para modificar el estado de un cuerpo basta alterar alguna de las acciones exteriores a que se halla sometido; por ejemplo, la presión, el campo magnético, etc. Aquellas magnitudes que pueden modificarse con independencia unas de otras entre todos los estados de un cuerpo, se denominan *variables de estado*."

"Se llaman *funciones de estado* aquellas magnitudes cuyo valor queda determinado en cuanto se atribuyen valores particulares a las variables de estado. Dicho de otro modo, son magnitudes cuyo valor depende del estado actual del cuerpo, pero no de sus estados anteriores."

"Las variables y las funciones de estado reciben el nombre genérico de *factores de estado*." Los párrafos entre comillas están tomados de [Palacios].

**Entonces el estado de un problema se describe con un conjunto de informaciones que utiliza las estructuras de datos habitualmente disponibles en las teorías matemáticas y en los lenguajes informáticos. Se trata de cadenas de símbolos, vectores, matrices, grafos, listas, colas, pilas, y en algunos casos, bases de datos que pueden definirse como estructuras compuestas de datos.**

Los sistemas pueden ser clasificados en estáticos y dinámicos.

El estado de un sistema dinámico varía con el tiempo.

Un sistema estático es aquel cuyo estado no varía con el tiempo.

Nótese que, dependiendo de la elección de las variables seleccionadas para representar el estado del sistema, una sección que la realidad que tiene apariencia de única puede ser considerada como estática por un observador y dinámica por otro.

Una estructura de construcción, por ejemplo, puede ser vista como un sistema estático (si factores tales como vibración, transferencia de calor con el entorno, dilataciones y contracciones pequeñas, etcétera se excluyen de los límites del sistema).

Un péndulo oscilante, un objeto caliente enfriándose, un sistema de tránsito con coches moviéndose, un piso de una fábrica o partes moviéndose, herramientas y trabajadores, un contexto de oficina con transacciones en papel, una red de comunicaciones con envío de mensajes y una computadora con transacciones de datos son varios ejemplos de sistemas dinámicos.

Los sistemas pueden ser clasificados aún más por la manera en que sus variables (que se han elegido para representar el estado del sistema) cambian a lo largo del tiempo.

Si estas variables del sistema cambian continuamente en el tiempo, el sistema es clasificado como un sistema continuo.

Si las variables del sistema cambian en forma discontinua a lo largo del tiempo, el sistema se clasifica como un sistema discreto.

Un objeto caliente que pierde su calor en el ambiente puede ser visto como un sistema continuo (si su estado es representado por la cantidad de calor que contiene) porque la cantidad de calor en el objeto es una variable cuyo valor cae continuamente al ir pasando el tiempo.

De la misma manera un lago detrás de una represa puede ser visto como un sistema continuo porque su nivel de agua varía continuamente al pasar el tiempo.

Este nivel es una variable que fluctúa por los efectos de las tasas de ingreso de los afluentes y las tasas de salida por medio de las turbinas de la represa, la evaporación del agua y el paso del agua a las napas freáticas.

Un banco que presta servicios a los clientes que arriban puede ser visto como un sistema discreto si el número de clientes en el banco representa el estado del sistema, y porque este número cambia sólo en cantidades distintas (nunca hay una fracción de cliente en el banco), entonces el estado del sistema cambia en forma discreta en el tiempo.

Aquí el estado del sistema es influenciado por el arribo y partida de clientes.

Nótese que ciertas variables (como los tiempos medios de espera del cliente) en este sistema del banco no son discretos, pero dado que no están en el conjunto de variables que representan el estado del sistema, el sistema aún es visto como discreto.

Si algunas variables cambian continuamente y otras cambian discretamente, al sistema se lo clasifica como **sistema combinado**.

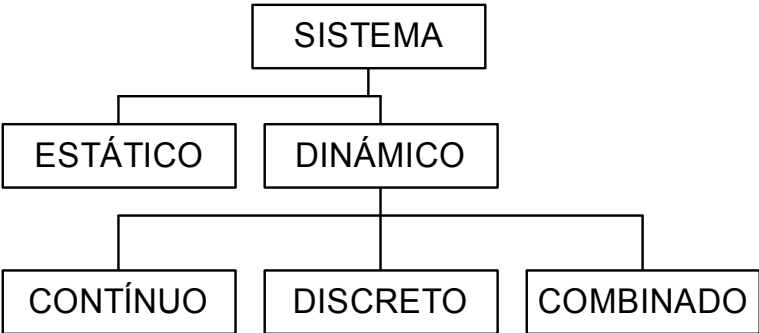
En el ejemplo de la represa, la salida total de agua cambia normalmente de forma continua a lo largo del tiempo.

El valor de la tasa de salida depende del nivel de presión del agua, la que a su vez depende del nivel de agua en el lago detrás de la represa.

Si algunas de las salidas de la represa son abiertas o cerradas instantáneamente, aparecerán cambios discretos en la tasa de salida del agua.

Si los sistemas son vistos como continuos o discretos, la variable tiempo en su análisis generalmente se considera continua (las excepciones son aquellas raras situaciones en las cuales los cambios discretos ocurren sólo en ciertos puntos del tiempo más que en cualquier otro punto). Todas las otras variables el sistema pueden depender de la variable tiempo, de otras variables o de parámetros del sistema.

La figura muestra la clasificación de sistemas que hemos hecho:



Recordemos nuevamente que la identificación del sistema y sus límites, así como la elección de verlo como estático, dinámico, continuo, discreto o combinado depende del propósito del analista al estudiar el sistema.

Una vez que se ha establecido el propósito del estudio, el conjunto de variables que representa el estado del sistema se elige cuidadosamente.

### Conceptos comunes de los sistemas dinámicos

Aunque sean vistos como continuos, discretos o combinados, todos los sistemas dinámicos siguen un comportamiento común.

Siempre hay un movimiento de algo físico (que puede incluir energía y medios de información).

Este algo físico se puede transformar en el curso de su movimiento. En los sistemas continuos nos referimos a este movimiento como flujo; en los sistemas discretos se le puede llamar movimiento.

El flujo o movimiento de lo físico ocurre a través de canales físicos que pueden (a causa de sus posibles limitaciones de capacidad) imponer restricciones en la tasa de flujo o movimiento.

Las mercaderías físicas se acumulan generalmente en ciertas partes del sistema.

Las cantidades acumuladas pueden ser agotadas por tasas de descarga.

En los sistemas continuos las cantidades acumuladas generalmente se llaman **niveles**; en los sistemas discretos se las llama **colas**.

Además del límite de capacidad del canal, el límite en el nivel máximo del punto de acumulación también pone una limitación natural en la tasa posible de salida de mercadería desde el punto de acumulación.

Las capacidades de los canales y de los puntos de acumulación se incluyen generalmente en el conjunto de parámetros del sistema (en ciertas situaciones puede ser variables).

Las tasas y cantidades de nivel puede ser consideradas variables dependientes del sistema que puede tomar valores diferentes en puntos diferentes del tiempo.

Para clarificar estos conceptos abstractos, veamos algunos ejemplos.

El agua en un sistema ecológico fluye constantemente a través de los canales de los ríos y las capas subterráneas y se acumula en lagos, mares, océanos y reservorios subterráneos.

El agua puede salir de esos reservorios aún en la forma de agua, a través de otros ríos y canales subterráneos para fluir en otros puntos de acumulación.

Influenciado por el calor del sol y del calor interno de la tierra, el agua se transforma en vapor de agua y sale a la atmósfera para acumularse en nubes.

Los límites de la tasa de salida y el límite de acumulación de vapor de agua son fijados por ciertos factores como la humedad atmosférica, temperatura y presión (nótese que las capacidades de los canales y de los puntos de acumulación puede ser variables dependientes en sí mismas más que parámetros fijos).

Como otro ejemplo, considere el movimiento de partes en el suelo de una fábrica. Las partes arriban a una cierta velocidad a una sección de maquinado.

Pueden entonces acumularse en un depósito de temporario antes de ser procesados.

La velocidad de procesamiento, la posibilidad de una rotura de la máquina, y la falta de disponibilidad de ciertas herramientas u operadores en la sección fijan una limitación en la tasa de partida de las partes.

Las partes que salen son transformadas en partes procesadas y puede continuar hacia otras secciones para otros procesos y ensamblados que resulta en una mayor transformación.

La mercadería puede tomar canales alternativos para fluir dentro y fuera de los puntos de acumulación. Dijimos que el flujo puede ser limitado y que esa limitación es generalmente fijada por la capacidad del canal y el valor del nivel que alimenta al canal.

En los sistemas hechos por el hombre, sin embargo, se pueden fijar ciertas limitaciones deliberadas en el flujo (o movimiento), la dirección del flujo, o las elecciones de ruteo a través de los canales alternativos del sistema.

Esta limitación impuesta deliberadamente es la esencia del control.

## Control

Es importante porque está relacionado a muchos análisis de sistemas y estudios de diseño.

Algunos de los problemas de control son los de control de tránsito, población, producción e inventarios.

El profesor supone que los alumnos dominan control.

## Características de los sistemas discretos

Vimos que los sistemas discretos se diferencian de los continuos en que su fuente de dinamismo está en forma disjunta (partes, autos, personas, etcétera) más que en conjunto (líquido, gas, calor, etcétera).

Nos referimos a cada parte discreta de la mercadería como a una **entidad**.

Entonces las entidades son unidades separadas cuyo movimiento en el sistema causa cambios discretos en el estado del mismo.

Las entidades no tiene porque ser mercaderías físicas.

Los mensajes y señales en un sistema de comunicaciones, por ejemplo, puede ser considerados como entidades.

Debe notarse que los sistemas discretos puede tener variable continuas.

Por ejemplo, la distancia entre las entidades autos es una variable continua.

Sin embargo, las variables continuas no aparecen en el conjunto de variables que caracterizan el estado del sistema discreto.

Las entidades pueden transformarse en otras entidades con características variadas (por ejemplo, una parte no procesada volviéndose una procesada).

También puede dividirse en un número mayor de entidades (por ejemplo la entidad ómnibus lleno transformándose en la terminal en la entidad ómnibus vacío y las entidades pasajeros).

Por último, varias entidades se pueden combinar para formar menos (por ejemplo, los componentes ensamblados que forman un producto).

Mencionamos que los cambios en el estado del sistema era causado por el movimiento de las entidades.

Esto no significa, sin embargo, que todo movimiento de entidades necesariamente resulta en un cambio del estado del sistema.

Para clarificar este punto, recuérdese que lo que representa el estado del sistema es relativo y depende del punto de vista del analista.

Por ejemplo si el estado del cuarto es representado sólo por el número personas en él, sólo puede cambiar (de forma discreta) cuando alguien entra o sale.

El movimiento de personas dentro del cuarto no afecta el estado prescrito del cuarto.

Este movimiento dentro del cuarto de las entidades son insignificantes desde el punto de vista del analista y, aunque ocurran en el mundo real, el estado del sistema real se asume constante y no afectado por su ocurrencia.

Como otro ejemplo supóngase que un analista está interesado en estudiar tanto el número de clientes dentro del banco y el número que está en una línea delante del escritorio del cajero.

El estado del sistema está representado por dos variables diferentes.

Cuando un cliente entra al banco, cambia el estado del sistema porque el número total de clientes en el banco cambia.

Mientras el cliente camina hacia la cola el estado del sistema es constante.

Una vez que el cliente llega al final de la línea el estado del sistema cambia otra vez dado que el número de clientes en cola es otra variable que representa el estado del sistema.

Nótese que en este ejemplo el estado del sistema cambia en ciertas instancias en el tiempo en el cual ocurren hechos significativos.

Las ocurrencias que resultan en cambios del estado de sistemas discretos se llaman **eventos**.

Las acciones que consumen tiempo pero que no incluyen el esperar en la cola (tal como el cliente caminando hacia la cola o recibiendo servicio en la ventanilla) generalmente se llaman **actividades** en los sistemas discretos.

Por tanto en los sistemas discretos el estado del sistema cambia sólo en tiempos de evento que tienen lugar al comienzo o al fin de actividades.

En el curso de sus actividades las entidades a veces necesitan usar y luego dejar (o consumir en la totalidad o en partes) ciertos **recursos** limitados.

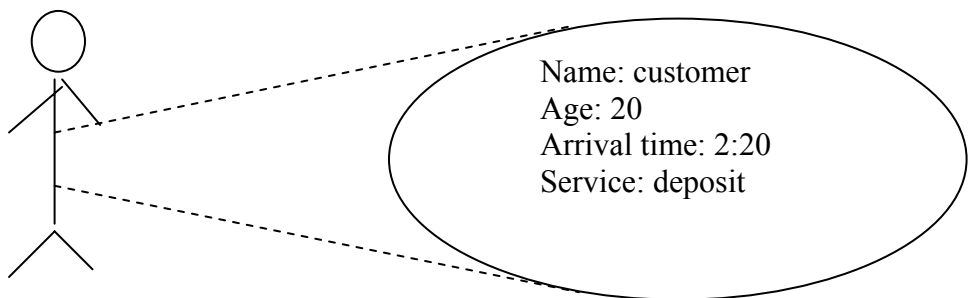
En estas situaciones las entidades pueden tener que esperar por su turno para usar los recursos.



Sin embargo, el esperar por recursos puede no ser la única demora que una entidad experimente. Por ejemplo, las entidades pueden tener que hacer una pausa y esperar permiso para moverse (como los autos esperando la luz verde). Las entidades pueden tomar diferentes rutas a través de varias actividades en las cuales puede tener que perder tiempos diferentes. La elección de las rutas que la entidad toma, el tiempo que gasta en actividades, la razón por la que espera, los recursos que usan y la forma en que los usan depende generalmente de ciertas condiciones. En nuestro ejemplo del banco, las entidades arribadas pueden elegir unirse a la cola más corta (si hay dos o más colas) y, dependiendo de si necesitan hacer un depósito o una extracción, sus servicios pueden tener diferentes tiempos de servicio. El reconocimiento de las entidades y sus rutas, la naturaleza de las actividades, los recursos del sistema, las varias condiciones que gobiernan el comportamiento del sistema, los hechos que llevan a eventos, y el impacto de los eventos en el estado del sistema son esenciales para el estudio de sistemas discretos.

### Más sobre entidades

Los sistemas discretos pueden tener varios tipos de entidades. Más, los mismos tipos de entidades pueden tener características variables. Por ejemplo, un sistema de transporte público puede incluir personas así como ómnibus que las cargan, llevan y descargan. Las personas tendrán características diferentes tales como su destino deseado y el monto que pueden gastar. De la misma forma, los ómnibus pueden tener diferentes capacidades, pueden ser asignados a diferentes rutas y pueden ser regulares o expresos con precios diferentes de los boletos. Las entidades pueden tener varias clases de relaciones unas con otras. Por ejemplo la entidad ómnibus puede transportar varias entidades pasajeros y una entidad policía puede parar una entidad ómnibus. Las entidades se caracterizan por sus atributos. La colección de atributos de una entidad se llama su **conjunto de atributos**. La figura muestra el conjunto típico de atributos de la entidad cliente de banco:



Nótese que cada atributo tiene un identificador (nombre, edad, tiempo de arribo, etcétera) y una instancia (cliente, 20, depósito, etcétera). El conjunto de atributos de una entidad es la información anexa a la entidad y encontrada con ella en su ubicación. Por lo tanto podemos considerar a los atributos como información local. Cierta otra información, tal como el tiempo, es global y puede ser observada desde cualquier lugar. Por ejemplo, en una cancha de fútbol la hora y el score de cada equipo puede ser exhibidos en un cartel que todos pueden ver (variables globales). Los jugadores individuales (entidades) llevan atributos tales como su número, el color de su equipo, el número de infracciones cometidas y cosas así que sólo tienen sentido cuando son expresadas respecto a un jugador particular. Entonces, al revés de lo que ocurre con la hora, el número de infracciones cometidas no pueden ser escritas en el tablero del estadio porque hay tantas de esas variables como jugadores. Las variables globales típicas utilizadas en los estudios de sistemas discretos son tiempo, conteo de entidades (número total de entidades de un cierto tipo que ha pasado a través de un cierto punto en el sistema), el número total de entidades en ciertos sectores del sistema (en una cola o en varias colas, actividades y rutas que conectan) y el estatus de varios recursos (número en uso, número disponible.) Para resumir la discusión acerca del comportamiento de sistemas dinámicos discretos, declaremos que **entidades** que poseen **atributos** y que están haciendo varias **actividades** y rutas se mueven a través del sistema, usan varios **recursos**, y crean **eventos** que cambian el **estado** del sistema, mientras mantienen ciertas **relaciones** lógicas.

Como una ayuda se incluyen las definiciones originales en inglés de [Khoshnevis].

*System.* A section of reality in the form of a set of components connected such that they can perform a function not performable by the individual components.

*Parameters.* Quantities in the system that do not change unless the analyst commands them to.

*Variables.* Quantities in the system that are determined by functional relationships and that change over time in dynamic systems.

*System state.* A snapshot of the system at any point in time characterized by the values of some selected variables in the system.

*Events.* Changes in the state of a discrete system.

*Entities.* Objects in the dynamic system whose motion within the system may result in the occurrence of events.

*Attributes.* Characteristics and properties that describe entities.

*Relationships.* Expressions of dependency between elements such as variables, parameters, and attributes of a system.

*Activities.* Time-consuming elements of a system whose starting and ending coincide with event occurrences.

*Resources.* Limited commodities that are used, consumed, or replenished by the entities.

*Control.* A mechanism that directs a dynamic system toward a set goal.

*Transient state.* Atypical conditions that impose time-dependent, radical perturbations on the system state.

*Steady state.* Typical conditions in which changes in the system state are within a fixed range and are independent of time.

Modelos

Para comunicar la descripción de un sistema o para analizar uno, primero debe expresarse el sistema con alguna forma de representación.

Esta representación se llama **modelo**.

Por ejemplo, un poeta puede tener un árbol (sistema) y representarlo en palabras que describen un árbol (representación verbal).

Un pintor puede mirar el mismo árbol pero expresarlo por medio de una pintura (representación pictórica).

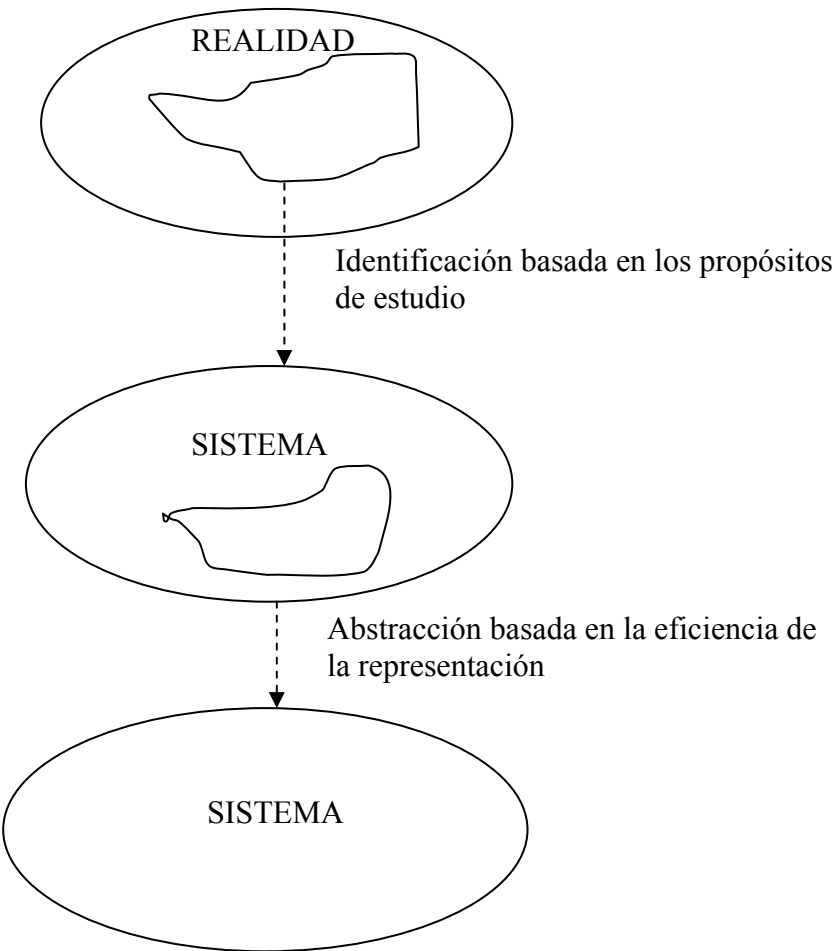
Aunque el poema y la pintura son diferentes, son modelos del mismo sistema.

Los modelos raramente cubren todos los hechos acerca del sistema que representan, en realidad son abstracciones de sistemas.

Ciertas formas de representación, sin embargo, son más eficientes para reflejar ciertos hechos.

Por ejemplo, la pintura de un árbol puede describir mejor los colores de la escena, pero el poema puede describir mejor el movimiento de las hojas y el canto de los pájaros.

La figura demuestra la relación entre realidad, sistema y modelo:



Nótese que lo que identifica al sistema es el propósito del estudio y una vez que es identificado lo que puede ser utilizado para identificar un modelo entre los modelos candidatos es la fuerza de la representación de las varias alternativas del modelado.

Los modelos no son sólo necesarios para la comunicación; son necesarios para análisis de sistemas.

Los ejemplos que vimos mostraban que los estudios de sistemas pueden tener el objetivo de entender el comportamiento de sistemas existentes o de hipotéticos sistemas futuros.

En general los estudios requieren experimentación y manipulaciones que son obviamente imposibles de implementar en los sistemas hipotéticos no existentes.

También, en el caso de sistemas existentes, esta experimentación y manipulación puede ser muy costosa de implementar en el propio sistema, dado que puede involucrar tareas costosas y que consumen tiempo.

También pueden causar distorsiones en la operación actual del sistema, y a veces pueden destruirlo.

Veamos unos ejemplos.

Si remodelamos el living-room, la experimentación con varios layouts cuando se implementa en el sistema en sí mismo requiere mover los objetos en el cuarto.

Tal experimentación, cuando se aplica al piso de una fábrica es aún más dramático dado que la maquinaria pesada debe ser sacada de sus fundaciones y movida para ver varias alternativas de layout.

Cuando intentamos encontrar el mejor timing de un semáforo para el cual el tiempo de espera de los autos que llegan a la intersección es mínimo, uno puede realmente manipular el timing del semáforo real por un cierto periodo de tiempo y observar el comportamiento del sistema bajo diferentes guiones de tiempo y esperar encontrar la secuencia de timing que resulte en la mejor performance del sistema.

Esta aproximación demanda la presencia del analista (y/o operadores y recolectores de datos) en el lugar del sistema por un lapso considerable y realmente puede distorsionar el flujo de tránsito en la intersección así como en las adyacentes.

Cuando se entrena y prueba la tripulación técnica de vuelo para ver si reacciona en la forma apropiada frente a situaciones de emergencia inesperadas (tal como falla del motor, turbulencia extrema, pérdida del control) el uso de un avión real en un vuelo real puede tener graves consecuencias como resultado de un error menor porque varias vidas y un aparato de varios millones de dólares pueden ser destruidos en una misión de entrenamiento.

Una circunstancia similar existe cuando se construye una represa (un sistema inexistente) con el fin de obtener irrigación y energía hidroeléctrica.

Si la construcción real se hace simultáneamente con el proceso de diseño, un proyecto multimillonario puede completarse después de varios años para encontrar que o hay una entrada insuficiente de agua o demasiada que en ciertas estaciones pasa encima de la presa e inunda las tierras agrícolas.

Para llegar al mejor diseño de represa utilizando esta aproximación, deberían construirse varias represas para observarlas en operación.

Los ejemplos sugieren que debe haber una forma mejor de hacer análisis de sistemas que experimentar con el sistema en sí mismo.

Los modelos sirven mejor a este fin.

**Un modelo es una representación del sistema que puede ser experimentado y manipulado con el fin último de estudiar al sistema.**

En esencia, los modelos se utilizan para hacer tres cosas:

1. **Estudiar sistemas existentes sin molestar su operación.**
2. **Estudiar sistemas existentes sin destruirlos.**
3. **Estudiar sistemas futuros inexistentes.**

Dado que se pueden utilizar varios esquemas cuando se representa un sistema, pueden representar al sistema muchas clases de modelos (que pueden ser diferentes).

Por lo tanto, un modelo válido puede o no parecerse al sistema que representa, pero debe contener todos los hechos importantes acerca del sistema.

Por ejemplo una circunferencia puede ser definida como un sistema de puntos que tienen la misma distancia, o radio, de un punto llamado el centro.

Una representación de este sistema estático es un modelo gráfico a la misma escala.

Otra representación puede ser una ecuación matemática que ubica el centro de la circunferencia en el origen de un par de ejes coordenados de dos dimensiones imaginario y especifica la circunferencia como todos los puntos donde el cuadrado de la coordenada X más el cuadrado de la ordenada Y iguala al cuadrado del radio.

## **Clasificación de los modelos**

Los modelos van desde réplicas físicas exactas del sistema hasta representaciones matemáticas abstractas.

Los modelos de sistemas pueden ser clasificados como **físicos**, **gráficos** o **simbólicos**.

Todos los tipos de sistemas (estáticos, dinámicos, discretos y combinados) pueden ser representados por algunos de los tipos de modelos.

Los modelos **físicos**, también llamados modelos **icónicos** pueden ser de la misma escala del propio sistema.



Un ejemplo de este tipo de modelo es la cabina de simulación utilizada para el entrenamiento del piloto. Otro es una planta operacional piloto construida para estudiar una nueva línea de producto anterior a la producción a escala completa.

Los modelos físicos pueden ser también de una escala menor del sistema que representan.

Ejemplos son la construcción a escala utilizada por los arquitectos y las maquetas de plantas químicas.

Algunos modelos a escala de sistemas tridimensionales pueden ser bidimensionales (tal como los modelos a escala del layout).

Modelos ampliados también son comunes como en los de cristales o estructuras de genes.

Los modelos físicos no tiene siempre que ser iguales al sistema que representan.

Por ejemplo, dado que los cambios de voltaje en un capacitor acoplado con un inductor sigue la misma pauta de intercambio de energía potencial que un peso oscilante colgado de un resorte, el sistema eléctrico es una analogía del sistema mecánico.

**Entonces un acoplamiento capacitor-inductor puede ser utilizado para modelar un resorte con un peso.**

**Este ejemplo sirve para mostrar que una característica común de los modelos físicos es que ellos son en sí mismos sistemas.**

Los modelos **gráficos** pueden ser representaciones de dos o tres dimensiones de sistemas.

Pueden ser estáticos, como los dibujos en papel, o dinámicos, como los filmes o los gráficos por computadora.

Las representaciones gráficas generalmente mejoran la comunicación y realzan la comprensión de modelos abstractos.

Por eso muchos modelos están soportados con representaciones gráficas e interfaces.

Por ejemplo, EZSIM, la herramienta de modelado simbólico con la cual se implementarán ejemplos, se beneficia de los gráficos.

Los modelos **simbólicos** son representaciones abstractas y como tales no se ven como los sistemas que representan.

En muchas aplicaciones estos modelos son un medio efectivo de representación del sistema por su facilidad de construcción y manipulación.

Por ejemplo una descripción verbal de un sistema es un modelo simbólico que puede ser expresado en cualquier lenguaje natural (inglés, chino, etcétera).

Un modelo verbal en ciertas aplicaciones es suficiente y mucho más fácil de construir que un modelo físico del sistema.

Siempre que se trate de un modelo sencillo.

Se requerirá mayor abstracción para expresar mejor las relaciones en el sistema bajo estudio.

Representaciones abstractas y organizadas de las reglas y hechos dentro del sistema tanto en la forma de procedimientos de lenguajes estructurados seminaturales, relaciones matemáticas y/o programas de computadoras son algunas de las formas usadas comúnmente de modelos simbólicos.

Como su nombre implica, los modelos simbólicos usan símbolos para representar componentes de sistemas (parámetros, variables, relaciones, etc.)

## Experimentación del modelo

El objetivo de construir un modelo no es sólo representar un sistema sino también experimentar con el modelo para varias configuraciones y valores de parámetros para tratar de obtener alguna conclusión útil independiente de la performance del sistema bajo estudio.

Para este fin el modelo debe ser evaluado —o, como se dice comúnmente, "resuelto"— para revelar el comportamiento del sistema.

En el caso de modelos físicos o gráficos, estas soluciones pueden ser fácilmente observadas y medidas.

En el caso de los modelos simbólicos, las soluciones pueden estar escondidas en la estructura abstracta del modelo.

Por ejemplo, en la representación gráfica del pronóstico de demanda para una empresa, la demanda pico y el momento de su ocurrencia puede ser fácilmente observada en el gráfico, pero cuando la misma relación de demanda con tiempo se representa matemáticamente (quizás como una función no lineal encontrada por ajuste de curvas que ajusten a los datos históricos), el "mirar" la representación de la demanda no revela mucha información acerca del punto máximo de la función.

En el último caso debe resolverse el modelo para obtener la información deseada.

Los métodos de solución para los modelos simbólicos son o analíticos o experimentales.

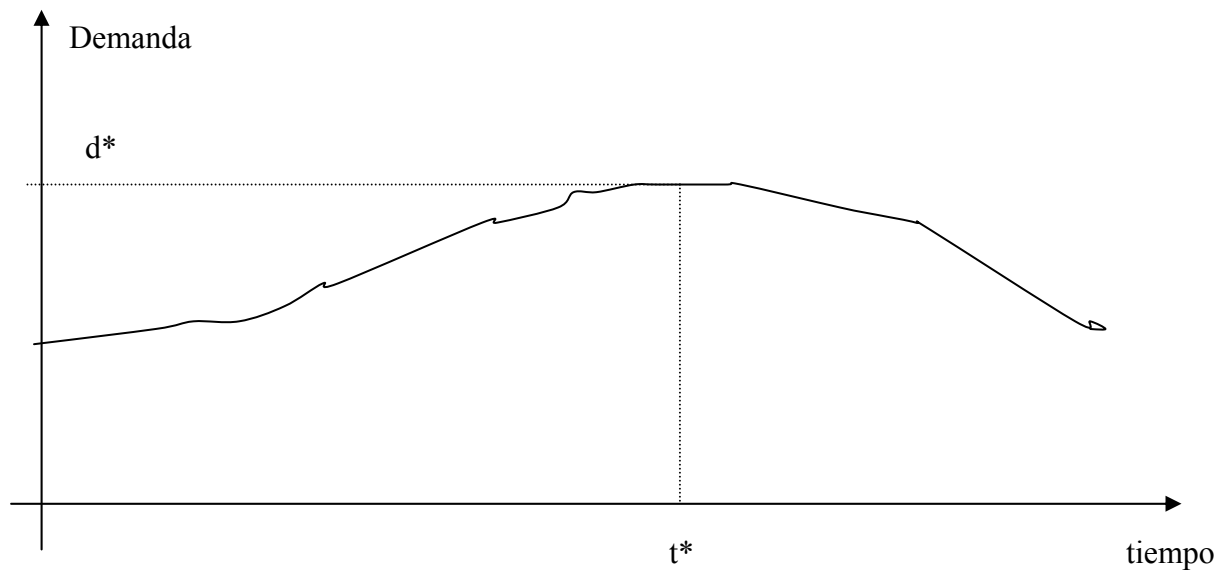
El método analítico se aplica generalmente a los modelos matemáticos; el método experimental se puede aplicar a los matemáticos así como a otras formas de modelos simbólicos.

El método analítico requiere el razonamiento deductivo de las teorías matemáticas que se aplican al problema que tenemos entre manos.

El método resulta generalmente en una solución que se obtiene rápido y con precisión.

Las soluciones analíticas son generales, pero un método experimental es específico y se aplica solo a un problema dado.

Como un ejemplo de aplicación del método analítico, considérese el problema de hallar el punto máximo del pronóstico de demanda que mencionamos.



Para hallar el máximo se iguala la derivada a cero y se resuelve la ecuación resultante para hallar el valor de tiempo en el máximo.

La mayor desventaja del método analítico para la resolución de problemas es que si el problema posee características no explicadas por la teoría matemática (un problema complejo) el analista debe desarrollar y probar nuevas teorías.

Dado que pocos analistas son matemáticos, puede ser que no conozcan las teorías que sirven para su problema, eso sin mencionar que sea capaz de establecer y probar nuevas teorías.

De cualquier manera, aún matemáticos experimentados son incapaces de encontrar soluciones analíticas atractivas para un amplio rango de problemas grandes y/o complejos.

El método experimental para la resolución de problemas generalmente utiliza aproximaciones simplificadas y a veces de sentido común más que teorías matemáticas sofisticadas.

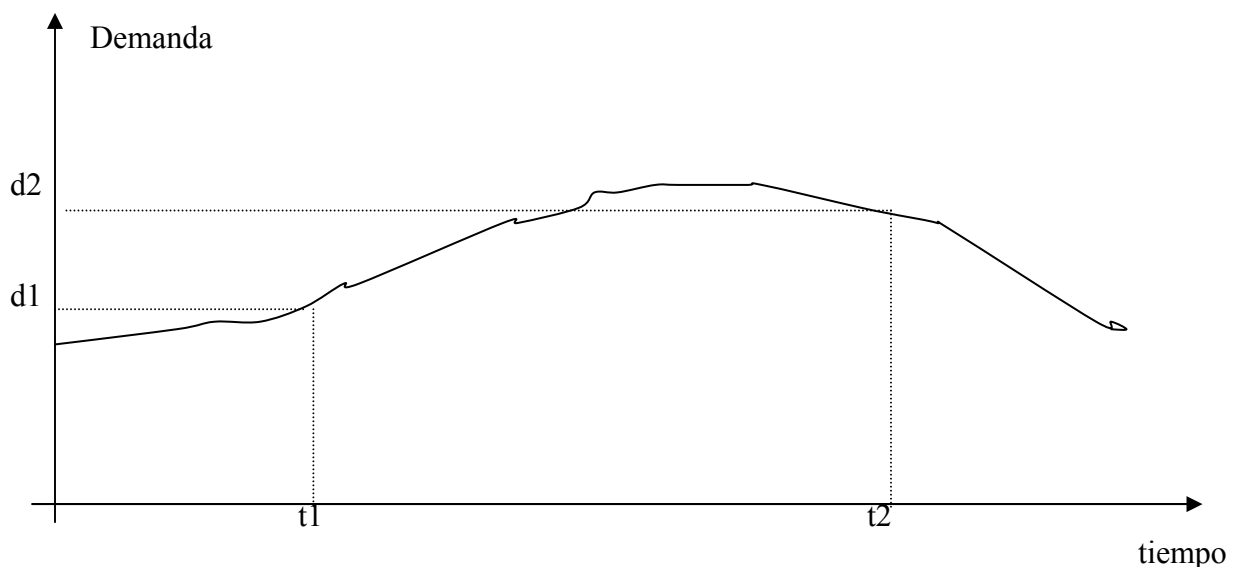
El precio por esta simplicidad y facilidad de uso es generalmente (no siempre) largos tiempos de computación y/o soluciones imprecisas.

Sin embargo, dado que la mayoría de los problemas del mundo real son complejos, a veces los métodos experimentales suelen ser las únicas alternativas factibles.

En lugar de elaborar la definición de los métodos experimentales, demostremos cómo pueden ser utilizados.

Dada la función de demanda en función del tiempo, para hallar el máximo sin ir a métodos matemáticos de optimización, podemos diseñar una aproximación simplista basada prueba y error.

Podemos comenzar con un punto arbitrario en el tiempo, digamos  $t_1$  y evaluar la función de demanda para este valor de tiempo.



Registramos este valor de demanda ( $d_1$ ) y tratamos con otro valor, digamos que  $d_2$ .

Entonces comparamos  $d_1$  y  $d_2$ . Si  $d_2$  es más grande que  $d_1$  podemos concluir que el máximo de demanda está más cerca de  $d_2$  que de  $d_1$ .

Subsecuentemente estudiamos el vecindario de búsqueda alrededor de que  $t_2$  y continuamos con el proceso de prueba hasta que dos puntos subsecuentes resultan en dos valores de demanda que son aproximadamente iguales entre sí (dentro de nuestro rango aceptable de error) y aceptamos el valor más grande como la cifra de la demanda máxima.

Cuando hallamos el área de un círculo el método experimental sería colocar algunos cuadrados adyacentes de igual tamaño dentro del círculo.

Contando el número de cuadrados y multiplicando este número por el área de uno de los cuadrados hallamos una aproximación al área del círculo.

Podemos reducir el error en esta aproximación eligiendo cuadrados más chicos, pero nuestro esfuerzo de computación se incrementará en forma acorde.

Otra forma experimental de hallar el área de un círculo es hacer un círculo en una pieza cuadrada de cartón de tamaño conocido, poner el cartón en una superficie plana y dejar caer pequeñas bolitas desde arriba de tal forma que las bolitas tengan una chance uniforme de caer en cualquier lugar de la superficie.

Entonces podemos tomar la proporción del número de bonitas que cae dentro del círculo y compararlo con el total de bolitas en el cartón.

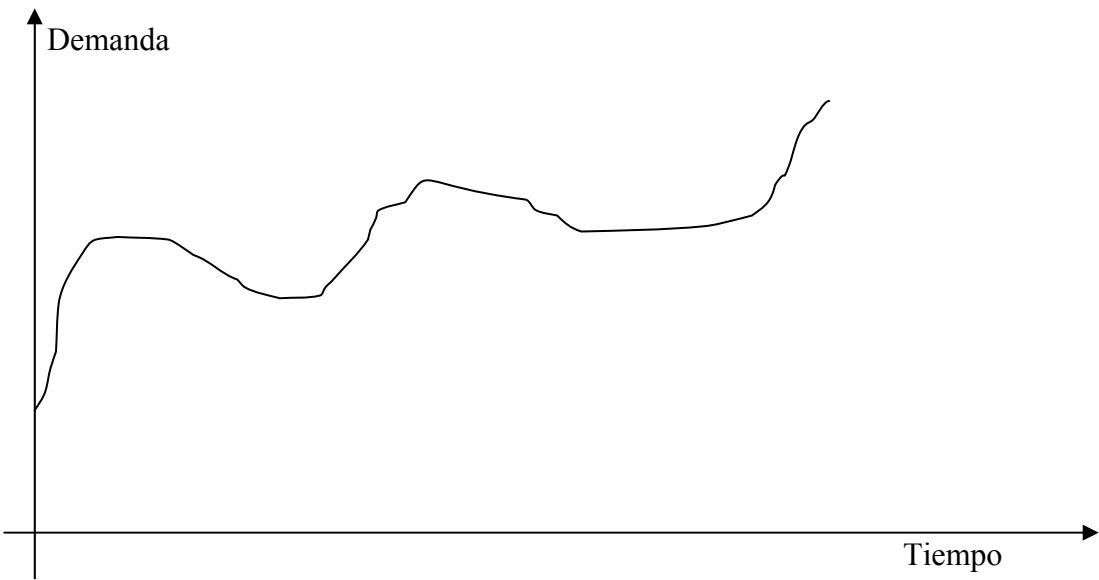
Ese número servirá como una estimación del área del círculo comparada con el área del cartón cuadrado.

La calidad de este estimado dependerá claramente del número de bolitas que dejamos caer.

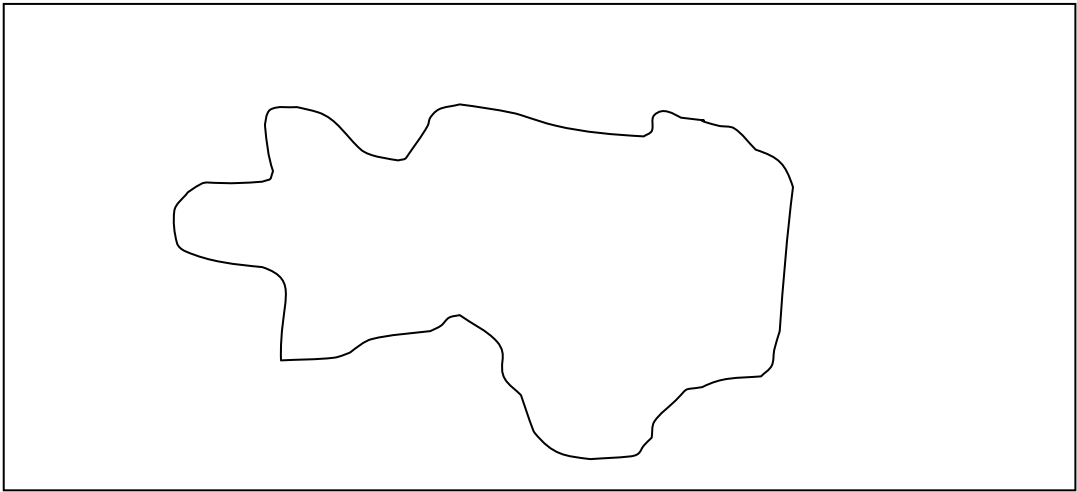
Si dejamos caer sólo una y esta cae fuera del círculo podemos concluir que el área del círculo es cero!

Esta aproximación está basada en la generación de muestras aleatorias y se llama la **técnica de Montecarlo**.

Para resaltar la potencialidad de los métodos experimentales consideremos una función demanda más realista como:



que es una demanda multimodal y



y un contorno no geométrico (tal como una foto aérea de un lago).

Aquí los métodos analíticos fallan para encontrar un máximo global (hay varios máximos) de la función porque resolver la primera derivada es una tarea compleja y ciertas funciones no tienen derivadas.

Tampoco podemos calcular analíticamente el área del lago.

O sea que los métodos experimentales que se han explicado más arriba trabajan bien en estos problemas complejos sin la necesidad de grandes modificaciones.

En el caso de la función demanda, el mismo algoritmo, si se intenta con varios puntos de inicio, encuentra el máximo global.

### Simulación — Una aproximación integral

Los ejemplos utilizados para demostrar los métodos analíticos y experimentales se referían a sistemas experimentales, o sea, no había movimiento de una mercadería.

Cuando nos enfrentamos a sistemas dinámicos, la estructura de los modelos que representan los sistemas generalmente se tornan más complejos.

En el caso de sistemas continuos, los métodos analíticos utilizados para experimentación de modelos utilizan técnicas de matemáticas avanzadas tales como ecuaciones diferenciales de varios órdenes que expresan cambios continuos de estado a lo largo del tiempo.

En el caso de sistemas discretos, especialmente cuando están presentes fenómenos estocásticos, son algo complicados de usar.

En tales casos se utilizan ecuaciones de transición de un estado discreto otro.

Las cadenas de Markov y la teoría de colas son métodos analíticos típicos utilizados en el campo de la investigación operativa para resolver este tipo de **sistemas dinámicos discretos**.

Sin embargo, esos modelos están restringidos a algunos modelos simples y limitados (a veces) al régimen estacionario (steady state).

Por las limitaciones de los métodos analíticos, la simulación ha probado ser una alternativa excelente en una variedad de problemas.

## **SIMULACIÓN**

Una intersección simple de tránsito con giro a la izquierda es imposible de estudiar con la teoría de colas salvo que se hagan simplificaciones no realistas.

Bajo esas condiciones el método experimental es la única aproximación viable.

### **Un ejemplo demostrativo de simulación**

El propósito de este estudio es encontrar el timing de los semáforos de una intersección que **minimice el tiempo promedio de espera de los autos que arriban a la intersección en un cierto período** (definición propósito).

Para evitar molestar la operación de la intersección real durante nuestro estudio, necesitamos construir un modelo del sistema.

Primero debemos observar el sistema real para identificar:

- Componentes básicos;
- la naturaleza de las tasas del sistema y
- la naturaleza de las relaciones entre los componentes.

Hagamos la hipótesis de que a través de la observación de la intersección encontramos que las calles que llegan a la intersección son ambas de doble mano.

También suponemos que los semáforos tienen flechas de giro a la izquierda y que se permite giros a la derecha sólo con luz verde.

Para re-crear qué es lo que ocurre en la realidad, debemos registrar la pauta de arribo de coches y su comportamiento una vez que arriban a la intersección (ir derecho, izquierda o derecha).

Para hacer esto podemos necesitar tener varios (digamos 4) asistentes, cada uno de ellos observando uno de los cuatro segmentos que llegan a la intersección.

Usando cronómetros estos asistentes registran el tiempo de arribo de cada coche (entidad) y la dirección a la que se dirige al salir de la intersección.

Dado que estamos tratando de encontrar un timing del semáforo mejor que el existente no habría necesidad de registrar el timing presente del semáforo (salvo que deseemos comparar la performance del nuevo timing del semáforo con la disposición actual).

Esta fase del estudio se denomina adquisición de datos.

La próxima fase del estudio es construcción del modelo.

Para facilitar la comprensión del concepto de simulación, usamos un modelo físico del sistema intersección.

Para construir el modelo físico usamos una placa plana, dibujar una intersección en ella y poner un semáforo miniatura en el centro.

Supongamos que el semáforo miniatura es realmente funcional y que es ajustado por un timer ajustable. Después tenemos que traer autitos a escala a la escena.

Ahora deberíamos llamar a los cuatro asistentes a escena y proveerles de algunos autitos.

Poniendo un reloj donde todos los asistentes puedan verle (el reloj master de simulación) les solicitamos que miren los tiempos que registraron las llegadas reales y hagan que sus autitos lleguen en un momento que coincida con el registrado.

También les pedimos que miren al semáforo de tal manera que sus autitos pasen a través de la intersección sólo si la luz correspondiente es verde.

De lo contrario deberán esperar detrás de la senda peatonal.

Al ir pasando el tiempo nosotros podemos ir manteniendo un registro de los tiempos de espera de los autitos.

Seguimos este proceso hasta que a nuestros asistentes se les terminen los datos (el final del periodo de simulación).

Entonces modificamos el timing del semáforo, reseteamos el reloj al principio del tiempo de la colección real de datos y repetimos el proceso completo.

Si nosotros y nuestros asistentes somos lo suficientemente pacientes, podemos repetir el proceso varias veces, cada vez con una disposición distinta del semáforo y para cada disposición registramos el tiempo de espera promedio de los autos (experimentación del modelo).

Nótese que en uno de los timings del experimento puede utilizar el correspondiente al sistema real.

Entonces las estadísticas cosechadas en el modelo pueden ser comparadas con las estadísticas de la performance del sistema real actual.

Si las dos estadísticas tienen resultados razonablemente próximos, tenemos una buena razón para creer que nuestro modelo es válido.

Utilizando la información registrada, finalmente podemos elegir el timing que corresponde al tiempo medio de espera más corto de los observados en nuestra simulación (análisis del output).

Entonces podemos ir a la intersección real y fijar el timing del semáforo real en el tiempo óptimo que hallamos en la simulación (implementación).

De esta manera nos hemos arreglado para estudiar el sistema y encontrar su mejor performance sin molestar la operación del semáforo durante el curso de nuestro estudio.

Nótese que los resultados de la simulación se aplican sólo al período durante el cual se ha hecho la adquisición de datos.

Un punto importante es que el reloj master de simulación no necesita ir a la misma de velocidad que el reloj real.

Por ejemplo, podemos hacer que nuestro reloj vaya a una velocidad diez veces más rápida que el tiempo real de reloj sin afectar la validez de nuestro estudio.

Esto es como si viéramos el sistema en cámara rápida (¡si los asistentes lo pueden resistir!).

Este ejemplo, aunque es bastante poco realista para implementar, muestra varios conceptos importantes acerca de la simulación.

**Primero:** ilustra el proceso de simulación que requiere de una definición de propósito, un estudio del sistema, adquisición de datos, construcción del modelo, validación del modelo, experimentación del modelo e implementación de los resultados.

**Segundo:** sirve para mostrar el concepto de tiempo de simulación y como éste está relacionado con el tiempo real: el ejemplo muestra que algunas horas en el sistema real (en su operación) pueden ser simuladas en pocos minutos.

Si los modelos están representados en forma simbólica y manipulados por computadoras, entonces el tiempo de simulación puede ser varias miles de veces más rápido que el real.

La simulación puede durar más que el tiempo real cuando simulamos sistemas extremadamente rápidos (por ejemplo movimiento de los electrones en la estructura molecular).

**Finalmente:** el ejemplo demuestra la dificultad involucrada en utilizar aproximaciones no simbólicas para estudiar algunos sistemas dinámicos.

## Un ejemplo numérico

Consideremos la terminal de service de una gran compañía transportista que posee varios aviones de carga.

Los aviones están programados para arribar a la terminal, uno al comienzo de cada día, para la posible operación de mantenimiento.

Al arribar, cada avión se inspecciona.

Supongamos que el tiempo de inspección es despreciable.

Una vez inspeccionados, la probabilidad de hallar que un avión necesita mantenimiento es de 0,5; o sea un promedio del 50% de los aviones necesita mantenimiento.

Si un avión necesita mantenimiento, esta operación puede requerir 0,5; 1; 1,5; 2; 2,5 ó 3 días.

La probabilidad de que el avión requiera uno de esos tiempos de servicio es de 1/6.

La compañía transportista utiliza un lugar de mantenimiento en la terminal.

Cada avión en tierra cuesta a la compañía 5.000 UM por día.

La gerencia de la compañía está interesada en investigar el atractivo económico de utilizar un lugar adicional de mantenimiento en la terminal.

Cada lugar le cuesta a la compañía 2.500 UM por día en leasing y operación.

El propósito del estudio, entonces, es comparar los costos operativos de los sistemas bajo los guiones de un lugar y dos lugares.

Nótese que en este ejemplo no se usan los detalles de datos registrados de los aviones que necesitan servicio ni de los tiempos de servicio.

En su lugar los datos se presentan en forma resumida.

O sea, los datos recopilados acerca de arribos de aviones y sus tiempos de service han sido analizados estadísticamente y resumidos como distribuciones estadísticas.

Hay ciertas ventajas en esta forma de representación de datos.

**Primero:** dada esta información sumaria, no estamos restringidos a un pequeño número de observaciones (como ocurría en el caso anterior).



Esto nos permite extender la longitud de la simulación y observar el comportamiento del modelo en lapsos mucho más grandes.

**Segundo:** podemos experimentar con varios parámetros de entrada (diferentes porcentajes de aviones que necesitan service y varias pautas de tiempos de servicio) verificar la performance del modelo en cada condición.

Pero esta forma de representación requiere un mecanismo para generar variables al azar que represente los comportamientos probabilísticos deseados.

Por ahora utilizamos un método simple para determinar qué aviones necesitan service y cuál es el tiempo de service requerido si se encuentra que un avión lo necesita.

El comportamiento probabilístico de nuestro sistema puede ser re-creado utilizando una moneda y un dado.

Estos simples **generadores físicos de variables aleatorias** (llamados a veces **generadores análogos**) simulan ocurrencias que tienen, respectivamente, probabilidades de ocurrir de 1/2 y 1/6.

Podemos simular la operación de la terminal de servicio tirando una moneda por cada arribo de un avión (lo que ocurre al comienzo del día) y si sale necesidad de mantenimiento, tiramos un dado para determinar cuánto tiempo tomará la operación de service.

Establezcamos que si la moneda cae de cara, asumiremos que el avión arribado necesita service.

Establezcamos que si el dado cae en 1, 2, 3, 4, 5 ó 6, la duración del servicio será 0,5; 1; 1,5; 2; 2,5; ó 3 días.

El resultado de una simulación de 10 días se muestra en la tabla.

Aquí, para cada día simulado de operación se tira una moneda.

Si es cara se tira un dado para saber el tiempo requerido de servicio.

Para entender la tabla es mejor concentrarse primero en las columnas relacionadas con un lugar y luego examinar el guión de dos lugares.

Nótese que en el guión de un lugar, algunos aviones que necesitan servicio deben esperar hasta que el lugar esté disponible.

El segundo guión, que usa dos lugares, es capaz de acomodar inmediatamente todos los aviones que necesitan service.

				Un lugar			Dos lugares			
DP	CT	DT	NDS	SSD	ESD	NDD	SSD	ESD	NDD	FN
0	Cara	5	2,5	0	2,5	2,5	0	2,5	2,5	1
1	Seca									
2	Cara	4	2	2,5	4,5	2,5	2	4	2	2
3	S									
4	C	6	3	4,5	7,5	3,5	4	7	3	1
5	C	3	1,5	7,5	9	4	5	6,5	1,5	2
6	S									
7	C	4	2	9	11	4	7	9	2	1
8	C	2	1	11	12	4	8	9	1	2
9	S									
Número de días perdidos						20,5			12	

Leyendas:

DP — days pass — número de día

CT — coin toss — tirada de moneda

DT — die toss — tirada de dado

NDS — number of days of service — número de días de servicio

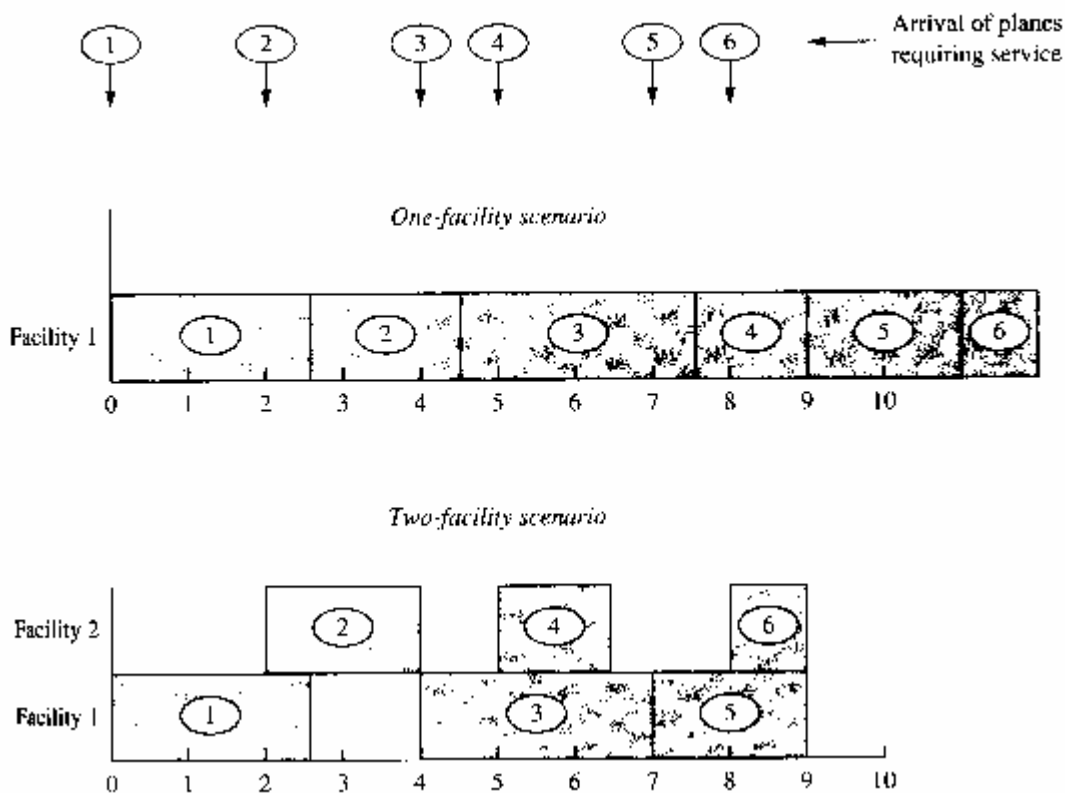
SSD — start service day — día de comienzo del servicio

ESD — end service day — día de finalización del servicio

NDD — number of days down — número de días en tierra

FN — facility number — lugar número

La figura muestra los mismos resultados en forma gráfica.



Basados en nuestro análisis de simulación el costo total de operación de la terminal de servicio será:

$$1 \text{ lugar} = 10 \cdot 2.500 + 20,5 \cdot 5.000 = 127.500 \text{ UM}$$

$$2 \text{ lugares} = 10 \cdot 5.000 + 12 \cdot 5.000 = 110.000 \text{ UM}$$

Esta simulación indica que utilizando dos lugares ahorrará a la compañía 17.500 UM por cada 10 días de operación (52.500 UM por mes).

Nótese que la confiabilidad en su validez depende de la fortaleza del experimento estadístico usando simulación.

Este ejemplo muestra que la simulación de sistemas estocásticos es en realidad un proceso de muestras. Cada repetición es una muestra de la operación del sistema.

Como ocurre para todos los métodos de muestreo, cuanto mayor sea el número de muestras tomado, mejor será la estimación del comportamiento del sistema.

Por ejemplo si en el modelo de la compañía transportista extendemos la longitud de la simulación a 100 días, podremos estar más confiados en el resultado de nuestro análisis.

De hecho, 10 días de observación es demasiado breve para obtener una muestra fiable.

Nótese, por ejemplo, que aunque nuestro experimento mostró que casi 50% de los aviones que arribaban necesitaban service (6 caras) hubiera sido probable obtener un resultado muy dispar de caras y secas (digamos 2 a 8), que no representa el comportamiento del sistema al largo plazo.

Esto es análogo al ejemplo comúnmente utilizado en estadística de tener un número de bolas blancas y negras en una bolsa.

Para estimar la proporción exacta de cada color de bola en la bolsa uno puede tomar una muestra.

Cuanto más grande sea el tamaño de la muestra, más cercana será la proporción de blancas y negras en la muestra a la proporción verdadera de cada bola en la bolsa.

Para evitar tediosas computaciones a mano elegimos una cantidad pequeña en la simulación de la operación de la compañía de transportes.

Si pudiéramos representar nuestra aproximación en la forma de un programa de computadora, seríamos capaces de simular el sistema por un lapso mucho mayor y obtener resultados mas confiables.

Por supuesto, debe haber un límite en la longitud de la simulación (tamaño de la muestra); de otra forma, el tiempo insumido en la corrida sería prohibitivo.

Esto es especialmente importante si el análisis requiere la aplicación de simulación día a día (como la generación de un programa de fabricación en una fábrica) más que una aplicación de una vez como en el caso de la compañía transportista.

## Representación por computadora de modelos de simulación

El libro de [Khoshnevis], en su apéndice A presenta el listado BASIC y en C de los programas utilizados por el software EZSIM.

No es necesario este programa dado que usaremos el software EZSIM que no necesita programación.

Creación de efectos aleatorios por computadora

En el ejemplo numérico de la compañía de fletes, instrumentos simples como una moneda o un dado, que tiene diferentes resultados (cara o sea; 1, 2, ..., 6) de probabilidades iguales de ocurrencia fueron utilizadas para recrear algunos efectos aleatorios en el modelo correspondiente de computadora. Hay otros artefactos físicos que pueden ser utilizados para crear ciertos resultados aleatorios continuos o discretos. Por ejemplo, una rueda de la fortuna bien balanceada con su borde numerados dentro de un rango. Cada tiro de la rueda resultará en una lectura en el indicador fijo, con un resultado uniformemente distribuido.

Ejemplo de simulación de evento discreto

La panadería Sud

La panadería Sud compra diariamente un cierto número de productos panificados. Se lleva stock de estos productos. Uno de los productos que compra Sud es un pan especial de trigo integral. Sud desea saber cuánto comprar diariamente porque desea maximizar las utilidades. Si compraba poco se perderían ventas y utilidades, si compraba mucho se desperdiciaría el exceso. Por simplicidad se supondrá que todo el pan no vendido se debe tirar al día siguiente, lo que implica una pérdida total. En la vida real quizá fuera posible vender los remanentes como pan de ayer con una pérdida parcial. Sud obtuvo datos sobre la demanda diaria de ese pan durante 100 días, obteniendo las siguientes frecuencias de demanda:

Demanda (Panes)	Punto medio	Frecuencia
20-24	22	0,05
25-29	27	0,10
30-34	32	0,20
35-39	37	0,30
40-44	42	0,20
45-49	47	0,10
50-54	52	0,05

Sud había administrado su inventario de pan comprando una cantidad igual a la demanda del día anterior. Esta regla de decisión, a veces, hacía que se quedara con demasiado pan y otras hacía que éste fuera insuficiente. Sud se preguntaba si existiría una regla mejor para la toma de decisiones. Por ejemplo, ¿qué sucedería con las utilidades si en lugar de comprar lo vendido en el día anterior se compraba el promedio?. En ese caso el promedio es:

$$\bar{X} = 0,05 \cdot 22 + 0,10 \cdot 27 + 0,20 \cdot 32 + 0,30 \cdot 37 + 0,20 \cdot 42 + 0,10 \cdot 47 + 0,05 \cdot 52 = 37$$

Para resolver este problema se construirá un modelo de simulación que incorpora las siguientes reglas de toma de decisiones:

- Regla 1: Ordene un número de panes igual a la demanda del día anterior.
- Regla 2: Ordene 37 panes por día, independientemente de la demanda previa.

Se seleccionará la regla que maximice las utilidades a lo largo de un número total de días simulados. Para ilustrar el método se tomarán sólo 15 días de simulación. En una aplicación real pueden utilizarse muchos más días para obtener resultados más confiables. Para simular este problema se generará una serie de demandas diarias aleatorias con frecuencias iguales a las que se dan antes. Para poder imaginar el procedimiento de generación de demanda, **imagínese una gran rueda de la fortuna con 100 posiciones**. Cinco de estas posiciones están marcadas con el número 22 que corresponde a la frecuencia antes mencionada de 0,05. Diez de las posiciones tienen el número 27 para indicar la frecuencia 0,10, y así sucesivamente. La rueda se hace girar una vez cada día y el lugar donde se detiene determina la demanda diaria. Como resultado, siempre y cuando la rueda sea verdaderamente aleatoria, se generarán demandas aleatorias de las frecuencias apropiadas. Aunque este proceso para la generación de demandas aleatorias es útil en forma conceptual, puede hacerse aún más eficiente utilizando una tabla de números aleatorios que puede encontrarse en cualquier libro de estadística.<sup>1</sup> La tabla puede utilizarse eligiendo un punto de arranque arbitrario y avanzando a

lo largo de filas o de columnas. Algunos de los dígitos pueden eliminarse si no son necesarios, debido a que los números en la tabla son completamente aleatorios.  
Por ejemplo, suponga que se desea generar 15 números aleatorios de dos dígitos para la panadería Sud. Suponga que se inicia en la esquina superior izquierda de la tabla (arbitrario) y se avanza a lo largo de la primera fila. Si se toman los dos primeros dígitos de cada número y se elimina el resto de los dígitos se obtendrá:

27, 43, 85, 88, 29, 69, 94, 64, 32, 48, 13, 14, 54, 15, 47

En este caso, podrían utilizarse también todos los dígitos de cada número o avanzar hacia abajo sobre las columnas en lugar de a lo largo de las filas o utilizar cualquier otro patrón que se desee, siempre que éste sea constante.  
Ahora que se han generado 15 números aleatorios, se les puede convertir en panes requeridos. Esto se hace relacionando el rango completo de 100 números aleatorios con la distribución de la demanda. En la tabla, el rango de 00 a 04 —un total de cinco números de los 100— se asigna a la primera categoría. En forma análoga, 05 a 14 contiene 10 números, es decir una frecuencia de 0,10. Queda una tabla como:

Punto medio de la demanda	Frecuencia	Números aleatorios
22	0,05	00 - 04
27	0,10	05 - 14
32	0,20	15 -34
37	0,30	35 - 64
42	0,20	65 - 84
47	0,10	85 -94
52	0,05	95 - 99

Los números aleatorios generados se convierten en los panes requeridos de la siguiente manera: el primer número aleatorio, 27, corresponde a una demanda de 32 panes; el segundo número, 43, corresponde a una demanda de 37 panes y así sucesivamente.  
De esta manera es posible simular la demanda de pan para cada uno de los 15 días del problema mediante lo que se denomina comúnmente el método Montecarlo.  
El paso siguiente será calcular las ventas correspondientes y la cantidad que se compra cada día. Estos cálculos dependen de la regla que se elija, por lo que se deberán hacer dos juegos de cálculos. Con la regla 1, la cantidad que se compra puede incluirse directamente. Las ventas serán entonces simplemente el mínimo entre la cantidad pedida y la cantidad demandada cada día. Sud no puede vender más de lo que compró.  
Se hacen cálculos similares para la regla 2. Los resultados se muestran en la tabla siguiente:

Panadería SUD						
			Regla 1		Regla 2	
Día	Número aleatorio	Demanda	Cantidad comprada	Ventas	Cantidad comprada	Ventas
0		37				
1	27	32	37	32	37	32
2	43	37	32	32	37	37
3	85	47	37	37	37	37
4	88	47	47	47	37	37
5	29	32	47	32	37	32
6	69	42	32	32	37	37
7	94	47	42	42	37	37
8	64	37	47	37	37	37
9	32	32	37	32	37	32
10	48	37	32	32	37	37
11	13	27	37	27	37	27
12	14	27	27	27	37	27
13	54	37	27	27	37	37
14	15	32	37	32	37	32
15	47	37	32	32	37	37

587	550	500	555	515
-----	-----	-----	-----	-----

Se muestran también las cifras de las cantidades compradas y vendidas en esos 15 días. Suponga que el pan se vende a 0,5 UM/u y cuesta 0,25 UM/u comprada al por mayor. Las utilidades totales a lo largo de 15 días serán:

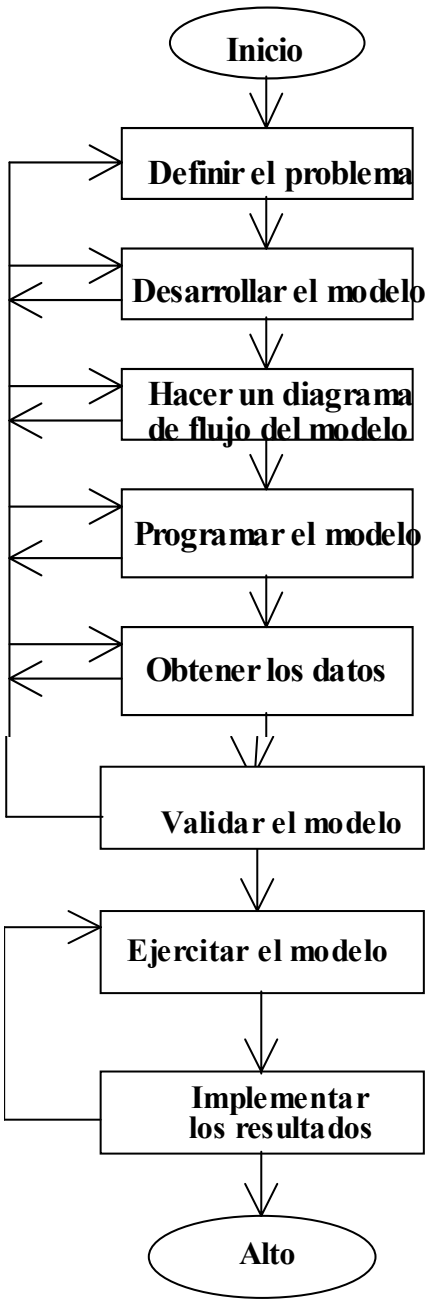
Regla 1:  $Utilidades = 0,5 \text{ UM/u} \cdot 500 \text{ u} - 0,25 \text{ UM/u} \cdot 550 \text{ u} = 112,5 \text{ UM}$

Regla 2:  $Utilidades = 0,5 \text{ UM/u} \cdot 515 \text{ u} - 0,25 \text{ UM/u} \cdot 555 \text{ u} = 118,75 \text{ UM}$

Como se ve, la regla 2 tiene un valor algo mayor de utilidad respecto a la regla 1.

### Metodología general

El ejemplo de la Panadería Sud representa una aplicación del método de simulación generalizado. Los pasos se muestran en la siguiente figura:



#### Definir el problema

Se requiere mucha experiencia e información para aislar un problema de su contexto con el fin de simularlo. La definición del modelo también supone las decisiones acerca de cuáles son los objetos, limitaciones e hipótesis que se utilizarán.

Después de definir el problema en términos generales, puede desarrollarse un modelo cuantitativo específico.

En la panadería Sud se aisló un problema relacionándolo con un tipo específico de pan. Su supuso que si no se disponía de un pan entero, se perdería la venta. No se consideró la posibilidad de ofrecer al cliente algún otro tipo de producto de panificación. También se supuso que el objetivo de Sud era maximizar las utilidades y así con gran cantidades de suposiciones explícitas e implícitas.

#### Desarrollo del modelo



En el desarrollo de modelos de simulación deben definirse:

- las variables controlables,
- las variables no controlables,
- las medidas del desempeño,
- las reglas de tomas de decisiones y
- las funciones del modelo.

De esta manera se desarrollará una representación matemática específica del problema. Todos los modelos pueden representarse de la siguiente forma:

$$P = f(U_i, C_k)$$

En el desarrollo del modelo de simulación simplemente se especifican los valores  $P, f, U_i$  y  $C_k$  del modelo.

En el modelo, las variables no controlables  $U_i$  son las constantes fijas no controlables que en SIMULACION se llaman datos. Por ejemplo, en la panadería Sud, el nivel inicial de la demanda, la frecuencia de distribución de la demanda, el costo de compra del pan y el precio de venta del mismo se consideraron como valores fijos.

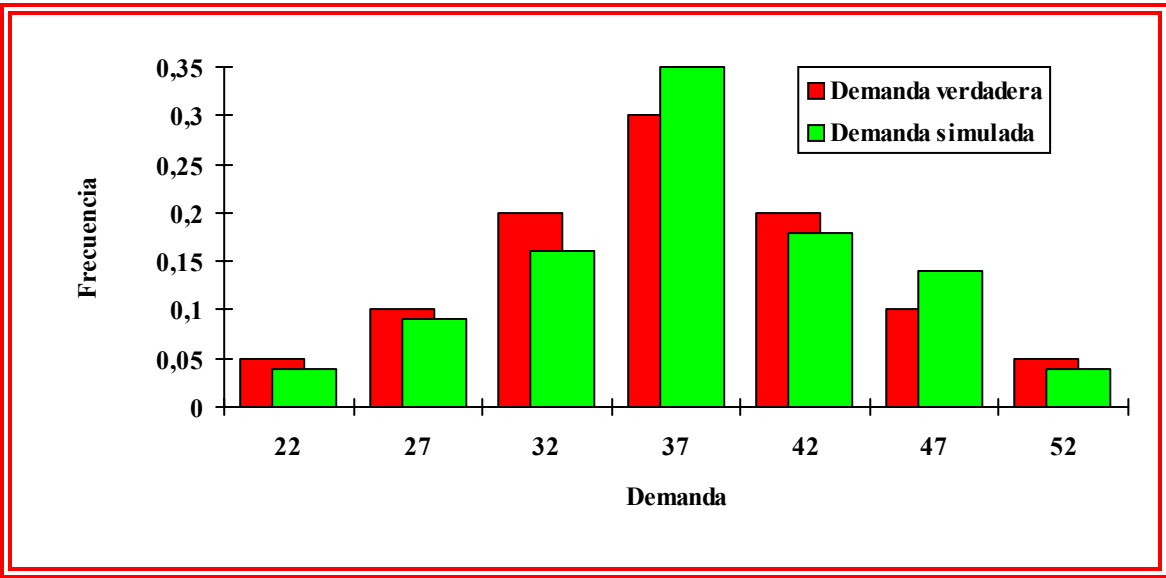
Las variables controlables  $C_k$  puede manejarlas quien toma la decisión. Estas variables pueden cambiar de valor de un ejercicio al siguiente conforme se simulan varias decisiones diferentes. En la panadería Sud, la cantidad de panes a comprar cada día es la variable controlable de la toma de decisiones. Los valores de esta variable se especificaron mediante reglas para la toma de decisiones que determinó la cantidad comprada cada día. Además, un modelo de toma de decisiones tiene una o más medidas de rendimiento  $P$  y funciones que relacionan las variables con el desempeño.

En la panadería Sud la medida del desempeño es la utilidad. Se utilizó una fórmula específica para calcular la utilidad como una función de las variables controlables y de los parámetros.

La validación de los parámetros busca determinar si los datos que se utilizaron en el modelo tiene relación con los valores correctos.

Por ejemplo, en el caso de la panadería Sud, debe probarse si la distribución de la demanda del pan generada por el modelo es una buena representación de una distribución supuesta.

Veamos cómo es la distribución de la demanda que se utilizó en el modelo versus los valores verdaderos de distribución de demanda:



En todos los valores de demanda hay diferencias entre los valores verdaderos y simulados.

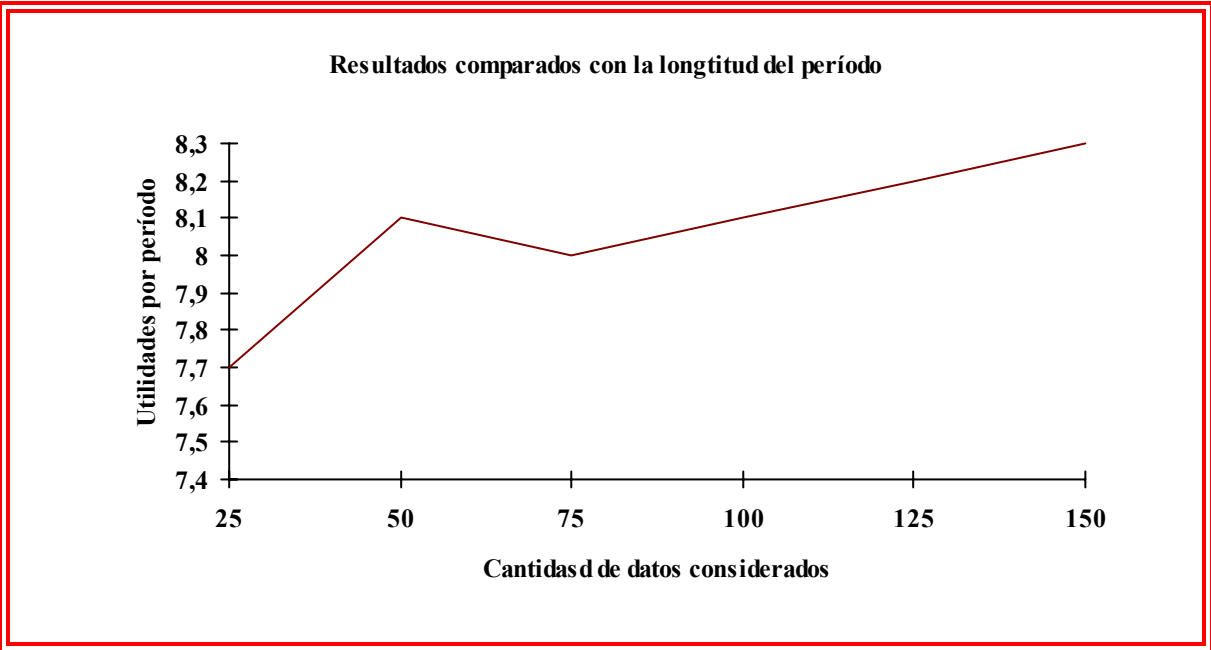
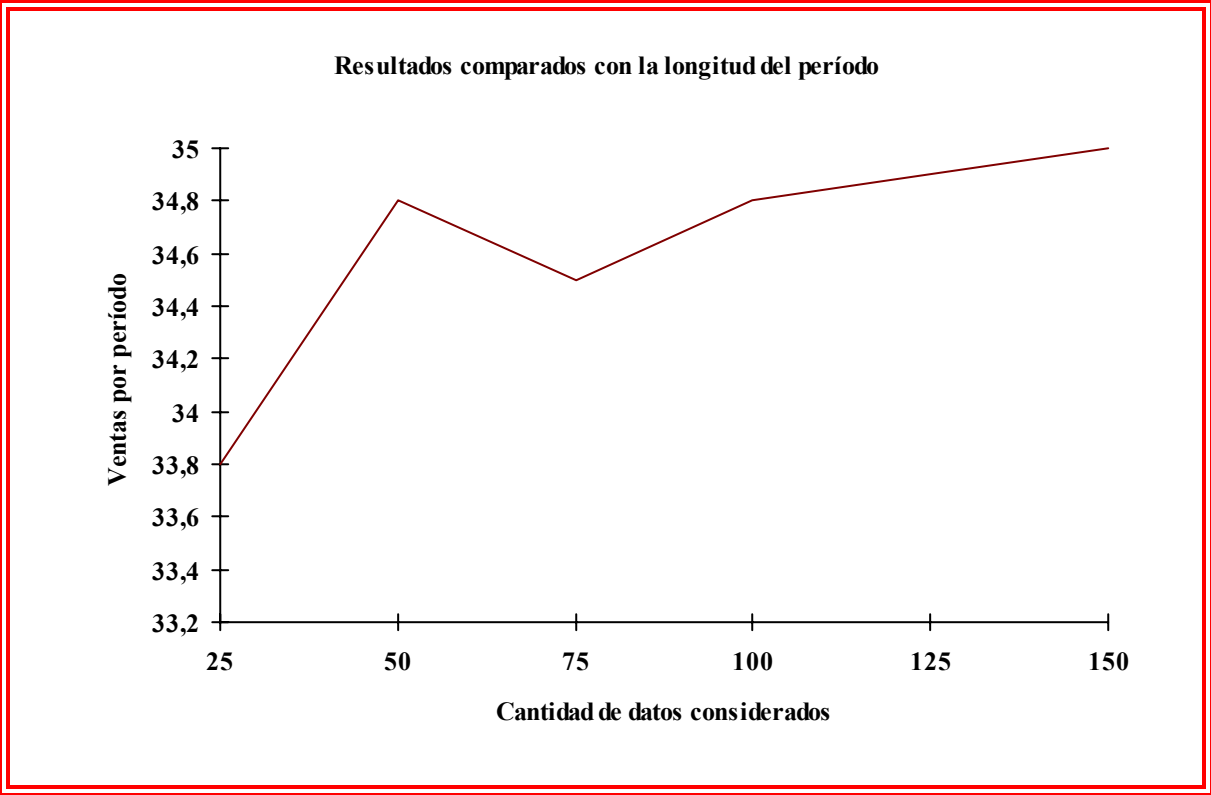
Existen técnicas estadísticas estándar para verificar si esas diferencias SON SIGNIFICATIVAS O NO.

Si las diferencias son significativas deben tomarse más datos o intentar otras hipótesis.

La validación puede también estar relacionada con la longitud del estudio.

En la simulación se establece un tamaño para el ejercicio con el fin de obtener resultados realistas.

Considérense los siguientes gráficos:



Estos gráficos muestran las ventas y las utilidades como una función de la cantidad de datos considerados.

Si los datos son pocos se ven fluctuaciones; sin embargo el resultado se estabiliza conforme se incrementa la longitud.

Esta validez debe revisarse, por eso en el diagrama de flujo se indicó la retroalimentación hacia los pasos anteriores.

Cuando la simulación esté validada, ya puede utilizarse el modelo.

En el caso de la panadería Sud, se hace un ejercicio para cada una de las dos reglas de decisión y se comparan los efectos sobre las ventas y las utilidades.

Se obtuvieron los siguientes resultados para 100 períodos:

	Regla 1	Regla 2
Utilidades	749 UM	810 UM
Ventas	1.687 UM	1.735 UM

En esta simulación se utiliza la misma demanda media diaria para ambas reglas, para que la variación de demanda sea la misma.

En los estudios de simulación se requiere un análisis de sensibilidad para probar cómo son sensibles los resultados contra las suposiciones de los datos de entrada.

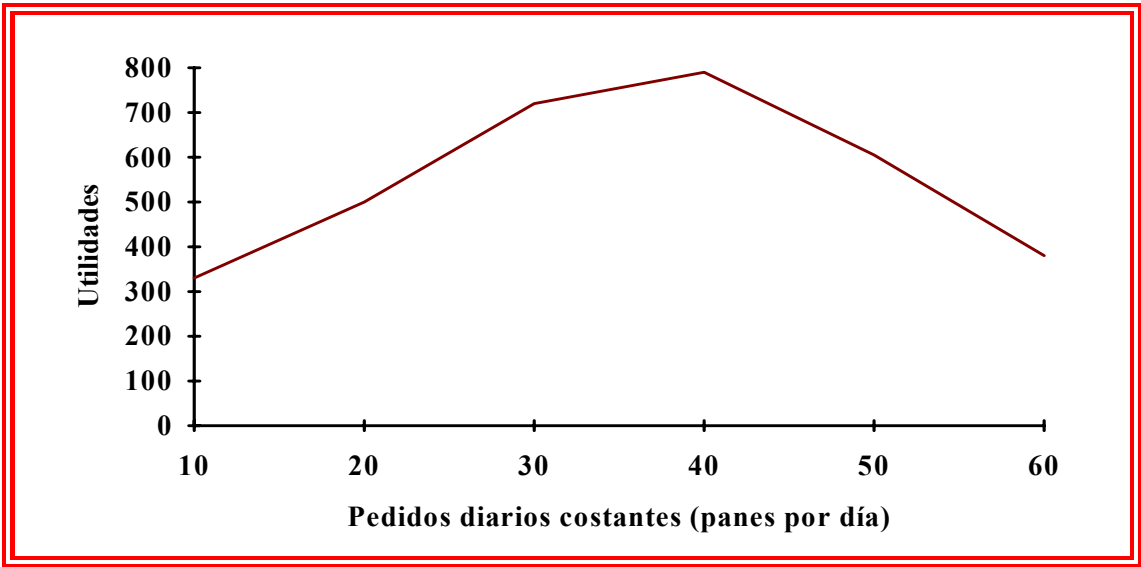
Por ejemplo, en el caso de panadería Sud, puede haber cierta falta de seguridad relacionada con el costo unitario.

En el ejemplo se utilizó un costo de 0,25 UM por pan.

Sin embargo, en realidad el costo podría ser de 30 ó de 20 centavos por pan, dependiendo de las hipótesis que se hicieran.  
¿Estos cambios en el costo afectarían la preferencia sobre la regla que se utiliza?  
Para poder contestar esta pregunta se hacen dos cálculos adicionales, uno donde  $C = 0,30$  y el otro donde  $C = 0,20$ .  
Los resultados de este cálculo son:

	C = 0,20		C = 0,30	
	Regla 1	Regla 2	Regla 1	Regla 2
	Utilidades	936 UM	561 UM	625 UM

Con base en las utilidades, se prefiere la regla 2 a la regla 1.  
Como parte del proceso de toma de decisiones, pueden realizarse ejercicios de sensibilidad similares seleccionándolos con los precios y con la demanda.  
En muchos casos, los ejercicios de sensibilidad son la parte más valiosa del análisis puesto que dan a la persona que decide el "sentir" de la situación.  
En el caso de la panadería Sud deben investigarse reglas de decisión adicionales.  
Por ejemplo, ¿por qué debe comprarse de acuerdo con el promedio de la demanda en el pasado? ¿No deben tenerse en consideración otros valores para los niveles de compra constantes?  
El siguiente gráfico:



muestra las utilidades como una función de varios niveles de compra constantes, indicando que resulta más rentable comprar 40 panes por día (lo cual es ligeramente mayor que el promedio de demanda previa).  
Como parte del análisis también deberían investigarse otras reglas.

TABLA 2

Lista de 500 números al azar, comprendidos entre 0 y 99999

22719	92549	10907	35994	63461	83659	24494	53825	97047	79069
17618	88357	52487	79816	74600	50436	88823	19806	33960	30928
25267	35973	80231	60039	50253	63457	97444	13799	35853	03149
88594	33679	66934	27705	51262	63941	77660	66418	84755	29197
60482	33679	03078	08047	39891	34068	81957	02985	83113	36981
30753	19458	02849	30366	83892	80912	91335	41703	79401	97251
60551	24788	35764	57453	06341	10178	91896	70819	46440	98356
35612	09972	98891	92625	70599	95484	34858	13499	28966	88287
43713	18448	45922	55179	18442	31186	91047	37949	76542	79361
73998	97374	66685	06639	34590	17935	79544	15475	74765	11199
14971	68806	49122	16124	61905	22047	17229	46703	39727	16753
78976	48382	25242	97656	51686	15537	73857	35398	91783	92825
37868	82946	73732	63230	85306	56988	15570	98029	42208	00190
01666	48114	95183	02628	05355	97627	75554	91267	31240	34723
56638	70054	19427	24811	37164	71641	50515	88231	99539	75745
43973	07496	17405	08966	65989	68017	56975	94080	93689	98889
05540	72301	36504	00187	90375	22891	22205	27777	84803	39220
95141	07885	94399	41145	50210	92423	13303	09621	94135	18691
75954	68499	42308	38387	52163	64563	02843	45577	93125	25294
97905	05301	98496	20682	68082	68537	70220	78282	02396	10002
23458	57782	67537	38813	00377	93873	97813	10039	25457	28716
03954	14799	63187	46191	12805	50502	08810	19572	48024	58206
52291	06804	85959	20974	73104	15009	25486	09306	24721	04187
62361	59105	39338	59358	69193	15586	57695	89518	59788	04215
54954	90337	99346	60442	90933	58323	83183	90041	44236	90815
70773	03331	84228	01405	61494	72064	24713	39851	01431	60841
68702	08331	08923	83173	67081	87472	47980	08802	95495	78745
39599	33465	96705	41458	34670	55385	25484	71068	15155	85371
54958	34935	16858	16523	54262	63310	50348	53457	39440	80411
98124	08864	36485	78766	52802	56315	43523	06513	50899	86423
43099	88373	80091	35058	35755	47556	98602	71744	70442	92312
88667	44515	80435	17140	32588	98708	93010	98590	23656	85664
87009	95736	76930	71090	27143	95229	24799	02313	17436	20273
70581	40618	16631	54178	44737	02544	81368	08078	46740	52583
03723	25551	03816	97612	99833	06779	47619	12901	60179	23780
49943	30139	07932	29267	01934	19584	13356	35803	90284	97565
71559	30728	83499	65977	37442	72526	53123	99948	59762	19952
75500	16143	79028	81790	57747	87972	54981	10079	17490	15215
59894	59543	13668	27197	51979	38403	23989	38549	82968	53300
29757	26942	08736	15184	73650	51130	59160	89966	06030	88929
87650	08162	90596	70312	84462	07653	80962	96692	07030	62470
84094	70059	86833	23531	31749	23930	04763	89322	67576	38627
92101	17194	06003	99847	12781	38729	88072	92589	61828	36504
26641	99088	65294	37138	75881	12627	19461	69536	64419	82106
04920	91233	46959	14735	15153	28306	76351	28109	86078	45234
25417	97570	91045	09929	75140	43926	90282	99088	93605	03547
98874	96989	84371	87624	74090	71983	62424	62130	44470	74725
82127	82000	84618	84618	58572	79862	59896	50702	31938	18336
26311	59516	98602	47197	31139	27631	64619	01504	77617	30219
76176	03499	17999	84361	63898	97861	63620	23931	87903	91566

Para hacer que la rueda sea útil para la creación de un resultado uniformemente distribuido dentro de un rango dado, el borde de la rueda puede ser numerado con un rango entre 0 y 1.

De esta manera, una simple transformación numérica de la lectura del indicador puede proveer el resultado distribuido aleatoriamente en forma uniforme.

Por ejemplo, si se necesitan números uniformemente distribuidos en el rango de 60 a 100, el indicador puede ser aumentado multiplicándolo por 40 (el rango) y elevando en 60, el extremo inferior del rango, para caer dentro del rango.

Por ejemplo, si el mantenimiento del avión se distribuye uniformemente con un rango entre 0,5 y 3 días la relación siguiente puede ser usada para crear la correspondiente duración:

$$x = 0,5 + r(3 - 0,5)$$

donde  $r$  es la salida aleatoria dada por la rueda y  $x$  es la variable que representa la duración aleatoria del mantenimiento, que esta distribuida entre 0,5 y 3 días.

Nótese también que sin el uso del rango 0—1 en la rueda y el método de transformación, podría necesitarse una numeración diferente de la rueda para cada resultado deseado uniformemente distribuido.

Los resultados que estén distribuidos uniformemente e independientemente dentro del rango 0—1 se llaman **números al azar**.

Dada su utilidad, los entornos de programación (por ejemplo BASIC, C y FORTRAN) ofrecen una función generadora de números al azar.

El siguiente programa BASIC genera y muestra 100 números aleatorios y puede ser usado para observar la performance de un generador de números aleatorios de una computadora típica:

```
10 FOR I = 100
20 PRINT RND (1)
30 NEXT I
```

La función RND ( ) utilizada es el generador de números aleatorios de BASIC y su argumento es la semilla para la corriente de números aleatorios generados por la función.

Si se cambia la semilla resulta una secuencia diferente de números aleatorios en una corrida diferente.

## Representación de la lógica del modelo por medio de programas de computadoras

Para representar el comportamiento de un sistema dinámico en un programa de computadora, deben ser conocidas las relaciones dentro y entre los componentes del sistema.

Este conocimiento determina la manera en que las variables del sistema deberían cambiar en el tiempo.

Dado este conocimiento adelantado (de acuerdo a cierto criterio) y para cada valor actualizado de tiempo los valores actualizados pueden ser computadas las variables del sistema.

De esta manera los cambios en el estado del sistema a lo largo del tiempo se pueden trazar y acopiar la estadística necesaria.

Dado que el tiempo de simulación juega un rol significativo en la determinación del flujo de control en un programa de simulación, el indicador de esto se llama usualmente reloj master de simulación (o reloj de simulación).

En la simulación de sistema continuo donde las variables pueden cambiar dentro de cualquier intervalo de tiempo, el tiempo de simulación usualmente es adelantado en pequeños incrementos y al final de cada incremento se computan nuevamente los valores de las variables que dependen del tiempo.

El tratamiento es diferente para sistemas discretos dado que en estos sistemas los valores de las variables no cambian entre dos eventos consecutivos.

Por lo tanto, el tiempo de circulación en sistemas discretos avanza de un evento a otro, y en cada tiempo de evento los valores de las variables afectadas por la ocurrencia del evento se actualizan.

**La clave para crear programas de computadora como modelos de simulación de sistemas discretos está en el reconocimiento de todos los eventos que pueden ocurrir y en el impacto que cada evento tiene en las variables del sistema.**

Generalmente los tiempos de ocurrencia de todos los eventos que suceden dentro del marco de tiempo de la simulación no se pueden conocer por adelantado.

**Esto es cierto porque los eventos usualmente son creados por otros que les preceden y de acuerdo con cierta lógica posiblemente complicada.**

Por ejemplo en un sistema de cola el evento que indica la salida desde una estación de atención depende de los eventos **tiempo de arribo, tiempo de servicio y estatus de la cola**.

O sea, aunque puede ser posible identificar por adelantado todos los elementos de arribos desde el inicio hasta el final de la simulación (dada la información acerca de los tiempos entre arribos), puede no ser tan simple predecir los eventos de partida por adelantado.

Debe señalarse que determinar todos los tiempos de eventos (tales como los tiempos de arribo) por adelantado y luego retroceder en el tiempo para encontrar los efectos de esos eventos en las variables del sistema no es una aproximación inteligente, dado que guardar información acerca de eventos que no ocurrirán en el futuro inmediato desperdicia memoria de computadora.



El manejo eficiente de eventos requiere una ventana limitada de tiempo que muestra los eventos más próximos.

La analogía de un auto moviéndose en la noche en una autopista con sus luces iluminando sólo una cierta distancia frente a él para ver las señales de tránsito y la ruta que viene sólo tan lejos como sea necesario para una conducción segura es apropiada.

En el trazado de la cadena de eventos en un programa de simulación, la aproximación debería identificar todos los efectos futuros inmediatos causados por el evento que está ocurriendo en ese momento de la simulación y debería ubicar esos eventos en una lista de eventos futuros.

El tiempo de simulación debe ser después avanzado al tiempo del evento más próximo en la lista y los cambios efectuados por el evento correspondiente deberían reflejarse en las variables del sistema.

Después de procesar cada evento, la información sobre este debe ser borrada de la memoria.

Para mostrar los temas más importantes en el desarrollo de un programa de simulación, consideremos un sistema simple: un guión de una cola simple de un canal con tiempos de arribo y de atención distribuidos uniformemente.

El programa de computadora asociado consiste en:

- la rutina de inicialización,
- la rutina de input,
- la rutina de timing del evento,
- la rutina del evento de partida,
- la rutina de estadísticas y
- la rutina de output.

Como ya dijimos, la realización de eventos y su efecto en las variables del sistema constituye un paso importante en el diseño de un programa de simulación de sistemas discretos.

En el ejemplo del sistema de cola, el archivo de la entidad, comienzo del servicio y partida de la entidad pueden ser considerados como eventos del sistema.

Entonces un módulo de programa se puede escribir para cada tipo de evento para hacer los cambios apropiados de las variables del sistema cuando ocurre un evento.

El reconocimiento de eventos interdependientes que ocurren al mismo tiempo puede reducir el número de módulos del programa relacionados con eventos.

El inicio de un evento de servicio en este caso es un ejemplo de evento dependiente porque ocurre cuando llega una entidad a un sistema vacío o al finalizar el servicio (partida) de una entidad cuando la cola no está vacía.

Por lo tanto la lógica del inicio de un servicio puede ser presentada en las rutinas de arribo y partida del programa de simulación.

El listado en BASIC no utiliza funciones especializadas y es simple; por lo que puede ser traducido a otros lenguajes como FORTRAN y PASCAL.

Se muestra un conjunto de parámetros de entrada y su correspondiente salida.

#### Input parameters

Lenght of simulation in time unis? 1.000  
Minimun interarrival time: 10  
Maximun interarrival time: 15  
Minimun service time? 9  
Maximun service time? 14

#### Simulation output

Number of entities entered = 81  
Number of entities served = 80  
Total time spent in queue = 93,68  
Average (expected) waiting time in queue = 1,17  
Average (expected) lenght at queue = 0,41  
Average server utilization = 57,91 %

Como se ha visto en el programa, un evento de arribo resulta siempre en programar un evento de arribo futuro a ocurrir en que TNA (time of next arrival).

Esto es conocido, dado que tenemos como datos los tiempos entre arribos.

En otras palabras, si conocemos el tiempo del arribo actual podemos determinar el momento del próximo arribo sacando un valor de la correspondiente distribución de probabilidad.

Un evento de arribo puede resultar en la programación de un evento de partida a TND (time next departure).

Esto ocurre sólo cuando una entidad que arriba encuentra desocupado al servidor.

Un evento de partida puede también programar otro elemento de partida.

Esto ocurre cuando un evento de partida ocurre desde un sistema con una cola no vacía.

De esta discusión se sigue que en nuestro sistema de cola de un canal puede haber un máximo de 2 de eventos inmediatos en la lista de eventos futuros, dado que hay siempre un evento de arribo y hay un evento de partida.

# PANADERIA SUD

OBJETIVO: Establecer una política de compras que mejore los beneficios.

Se simulan 200 eventos evaluando 3 políticas o reglas.

Regla N° 1: Se compra en función de la demanda del día anterior.

Regla N° 2: Se compra de acuerdo al promedio histórico de la demanda.

Regla N°. 3: Se va corrigiendo la compra en función de la diferencia entre el promedio y el promedio de los sucesos anteriores.

TABLA DE FRECUENCIAS

NUM ALEAT	DEMANDA	FRECUENCIA
0	22	0,05
5	27	0,10
15	32	0,20
35	37	0,30
65	42	0,20
85	47	0,10
95	52	0,05

Días	Generación de número aleatorio	Números aleatorios 0—99	Demanda panadería según distribución	Regla 1		Regla 2		Regla 3	
				Com.	Ven.	C	V	C	V
0	0,2862092	28	32	37	32	37	32	40	32
1	0,5596681	55	37	32	32	37	37	38	37
2	0,2337290	23	32	37	32	37	32	39	32
3	0,6340731	63	37	32	32	37	37	38	37
4	0,3060794	30	32	37	32	37	32	39	32
5	0,2138854	21	32	32	32	37	32	39	32
6	0,3234285	32	32	32	32	37	32	39	32
7	0,0691855	6	27	32	27	37	27	39	27
8	0,7213629	72	42	27	27	37	37	39	37
9	0,3027253	30	32	42	32	37	32	39	32
10	0,6088553	60	37	32	32	37	37	39	37
11	0,4802340	48	37	37	37	37	37	38	37
12	0,4933226	49	37	37	37	37	37	38	37
13	0,3804361	38	37	37	37	37	37	38	37
14	0,2948388	29	32	37	32	37	32	38	32
15	0,2682578	26	32	32	32	37	32	38	32
16	0,1217963	12	27	32	27	37	27	39	27
17	0,7633742	76	42	27	27	37	37	38	37
18	0,6347764	63	37	42	37	37	37	38	37
19	0,1410233	14	27	37	27	37	27	39	27
20	0,8833362	88	47	27	27	37	37	38	37
21	0,0653906	6	27	47	27	37	27	38	27
22	0,9825311	98	52	27	27	37	37	38	37
23	0,2547113	25	32	52	32	37	32	38	32
24	0,2391606	23	32	32	32	37	32	38	32
25	0,7071177	70	42	32	32	37	37	38	37
26	0,6199481	61	37	42	37	37	37	38	37
27	0,5039552	50	37	37	37	37	37	38	37
28	0,8376556	83	42	37	37	37	37	38	37
29	0,6652986	66	42	42	42	37	37	38	37
30	0,3789865	37	37	42	37	37	37	38	37
31	0,3732208	37	37	37	37	37	37	38	37
32	0,5725422	57	37	37	37	37	37	38	37
33	0,9144980	91	47	37	37	37	37	37	37
34	0,8585769	85	47	47	47	37	37	37	37
35	0,2383811	23	32	47	32	37	32	37	32
36	0,8373314	83	42	32	32	37	37	37	37
37	0,8893219	88	47	42	42	37	37	37	37
38	0,0018309	0	22	47	22	37	22	37	22
39	0,5716109	57	37	22	22	37	37	37	37
40	0,3666940	36	37	37	37	37	37	37	37

41	0,5452691	54	37	37	37	37	37	37	37
42	0,0492273	4	22	37	22	37	22	37	22
43	0,7527275	75	42	22	22	37	37	37	37
44	0,3053695	30	32	42	32	37	32	37	32
45	0,8463415	84	42	32	32	37	37	37	37
46	0,1333996	13	27	42	27	37	27	37	27
47	0,4401011	44	37	27	27	37	37	37	37
48	0,3952386	39	37	37	37	37	37	37	37
49	0,7575366	75	42	37	37	37	37	37	37
50	0,0311500	3	22	42	22	37	22	38	22
51	0,7042609	70	42	22	22	37	37	37	37
52	0,4211848	42	37	42	37	37	37	37	37
53	0,0703482	7	27	37	27	37	27	38	27
54	0,8383155	83	42	27	27	37	37	38	37
55	0,3773214	37	37	42	37	37	37	37	37
56	0,3063710	30	32	37	32	37	32	38	32
57	0,6393993	63	37	32	32	37	37	38	37
58	0,5831096	58	37	37	37	37	37	38	37
59	0,3213539	32	32	37	32	37	32	38	32
60	0,4975538	49	37	32	32	37	37	38	37
61	0,8583335	85	47	37	37	37	37	37	37
62	0,0435670	4	22	47	22	37	22	38	22
63	0,5688283	56	37	22	22	37	37	38	37
64	0,5310626	53	37	37	37	37	37	38	37
65	0,4530888	45	37	37	37	37	37	38	37
66	0,2847132	28	32	37	32	37	32	38	32
67	0,5382506	53	37	32	32	37	37	38	37
68	0,9200680	92	47	37	37	37	37	37	37
69	0,9206796	92	47	47	47	37	37	37	37
70	0,6810868	68	42	47	42	37	37	37	37
71	0,7781201	77	42	42	42	37	37	37	37
72	0,7537635	75	42	42	42	37	37	37	37
73	0,1916110	19	32	42	32	37	32	37	32
74	0,4402382	44	37	32	32	37	37	37	37
75	0,9423324	94	47	37	37	37	37	37	37
76	0,6490870	64	37	47	37	37	37	37	37
77	0,3440510	34	32	37	32	37	32	37	32
78	0,8928293	89	47	32	32	37	37	37	37
79	0,9501526	95	52	47	47	37	37	37	37
80	0,9456059	94	47	52	47	37	37	37	37
81	0,4478861	44	37	47	37	37	37	37	37
82	0,0649605	6	27	37	27	37	27	37	27
83	0,5338836	53	37	27	27	37	37	37	37
84	0,2841915	28	32	37	32	37	32	37	32
85	0,8667840	86	47	32	32	37	37	37	37
86	0,2983488	29	32	47	32	37	32	37	32
87	0,6273548	62	37	32	32	37	37	37	37
88	0,9601074	96	52	37	37	37	37	37	37
89	0,5963896	59	37	52	37	37	37	37	37
90	0,9339827	93	47	37	37	37	37	37	37
91	0,2983567	29	32	47	32	37	32	37	32
92	0,6911367	69	42	32	32	37	37	37	37
93	0,4667784	46	37	42	37	37	37	37	37
94	0,3248783	32	32	37	32	37	32	37	32
95	0,1781445	17	32	32	32	37	32	37	32
96	0,6913739	69	42	32	32	37	37	37	37
97	0,2872929	28	32	42	32	37	32	37	32
98	0,6935623	69	42	32	32	37	37	37	37
99	0,5911384	59	37	42	37	37	37	37	37
100	0,1913050	19	32	37	32	37	32	37	32
101	0,2618871	26	32	32	32	37	32	37	32
102	0,5999067	59	37	32	32	37	37	37	37
103	0,8379437	83	42	37	37	37	37	37	37
104	0,7948027	79	42	42	42	37	37	37	37
105	0,2531035	25	32	42	32	37	32	37	32

106	0,0349395	3	22	32	22	37	22	37	22
107	0,3348153	33	32	22	22	37	32	37	32
108	0,3136596	31	32	32	32	37	32	37	32
109	0,7228077	72	42	32	32	37	37	37	37
110	0,4062577	40	37	42	37	37	37	37	37
111	0,5626946	56	37	37	37	37	37	37	37
112	0,9702861	97	52	37	37	37	37	37	37
113	0,4278063	42	37	52	37	37	37	37	37
114	0,9306658	93	47	37	37	37	37	37	37
115	0,0289392	2	22	47	22	37	22	37	22
116	0,1091972	10	27	22	22	37	27	37	27
117	0,1984724	19	32	27	27	37	32	37	32
118	0,1996673	19	32	32	32	37	32	37	32
119	0,6948750	69	42	32	32	37	37	37	37
120	0,8079136	80	42	42	42	37	37	37	37
121	0,3785037	37	37	42	37	37	37	37	37
122	0,9334315	93	47	37	37	37	37	37	37
123	0,5722926	57	37	47	37	37	37	37	37
124	0,0279014	2	22	37	22	37	22	37	22
125	0,7539390	75	42	22	22	37	37	37	37
126	0,1888626	18	32	42	32	37	32	37	32
127	0,4420140	44	37	32	32	37	37	37	37
128	0,1961761	19	32	37	32	37	32	37	32
129	0,0495565	4	22	32	22	37	22	37	22
130	0,5373946	53	37	22	22	37	37	37	37
131	0,8976550	89	47	37	37	37	37	37	37
132	0,3581958	35	37	47	37	37	37	37	37
133	0,3207813	32	32	37	32	37	32	37	32
134	0,9781391	97	52	32	32	37	37	37	37
135	0,1677076	16	32	52	32	37	32	37	32
136	0,2926682	29	32	32	32	37	32	37	32
137	0,8192242	81	42	32	32	37	37	37	37
138	0,6020082	60	37	42	37	37	37	37	37
139	0,4190679	41	37	37	37	37	37	37	37
140	0,2288969	22	32	37	32	37	32	37	32
141	0,1599799	15	32	32	32	37	32	37	32
142	0,2724397	27	32	32	32	37	32	37	32
143	0,4951342	49	37	32	32	37	37	37	37
144	0,0572917	5	27	37	27	37	27	37	27
145	0,8688062	86	47	27	27	37	37	37	37
146	0,8383490	83	42	47	42	37	37	37	37
147	0,2916005	29	32	42	32	37	32	37	32
148	0,1736676	17	32	32	32	37	32	37	32
149	0,6373128	63	37	32	32	37	37	37	37
150	0,6449321	64	37	37	37	37	37	37	37
151	0,2230704	22	32	37	32	37	32	37	32
152	0,3479587	34	32	32	32	37	32	37	32
153	0,8642817	86	47	32	32	37	37	37	37
154	0,3579454	35	37	47	37	37	37	37	37
155	0,4836158	48	37	37	37	37	37	37	37
156	0,2579700	25	32	37	32	37	32	37	32
157	0,3112460	31	32	32	32	37	32	37	32
158	0,3670873	36	37	32	32	37	37	37	37
159	0,0544067	5	27	37	27	37	27	37	27
160	0,5787945	57	37	27	27	37	37	37	37
161	0,9504510	95	52	37	37	37	37	37	37
162	0,2841057	28	32	52	32	37	32	37	32
163	0,6615356	66	42	32	32	37	37	37	37
164	0,7685186	76	42	42	42	37	37	37	37
165	0,3333240	33	32	42	32	37	32	37	32
166	0,3012577	30	32	32	32	37	32	37	32
167	0,1371629	13	27	32	27	37	27	37	27
168	0,9408413	94	47	27	27	37	37	37	37
169	0,0575757	5	27	47	27	37	27	37	27
170	0,4257459	42	37	27	27	37	37	37	37

171	0,8871066	88	47	37	37	37	37	37	37
172	0,4652335	46	37	47	37	37	37	37	37
173	0,1610606	16	32	37	32	37	32	37	32
174	0,8557552	85	47	32	32	37	37	37	37
175	0,4180037	41	37	47	37	37	37	37	37
176	0,1852605	18	32	37	32	37	32	37	32
177	0,5205194	52	37	32	32	37	37	37	37
178	0,0137181	1	22	37	22	37	22	37	22
179	0,1271655	12	27	22	22	37	27	37	27
180	0,3956936	39	37	27	27	37	37	37	37
181	0,5833295	58	37	37	37	37	37	37	37
182	0,0024292	0	22	37	22	37	22	37	22
183	0,8082546	80	42	22	22	37	37	37	37
184	0,9690364	96	52	42	42	37	37	37	37
185	0,0103885	1	22	52	22	37	22	37	22
186	0,4996386	49	37	22	22	37	37	37	37
187	0,5759054	57	37	37	37	37	37	37	37
188	0,2720895	27	32	37	32	37	32	37	32
189	0,0582550	5	27	32	27	37	27	37	27
190	0,3567449	35	37	27	27	37	37	37	37
191	0,1624173	16	32	37	32	37	32	37	32
192	0,7631794	76	42	32	32	37	37	37	37
193	0,7786727	77	42	42	42	37	37	37	37
194	0,4172160	41	37	42	37	37	37	37	37
195	0,7896407	78	42	37	37	37	37	37	37
196	0,4106767	41	37	42	37	37	37	37	37
197	0,3636021	36	37	37	37	37	37	37	37
198	0,3203612	32	32	37	32	37	32	37	32
199	0,7549152	75	42	32	32	37	37	37	37
200				7295	6500	7400	6835	7478	6835

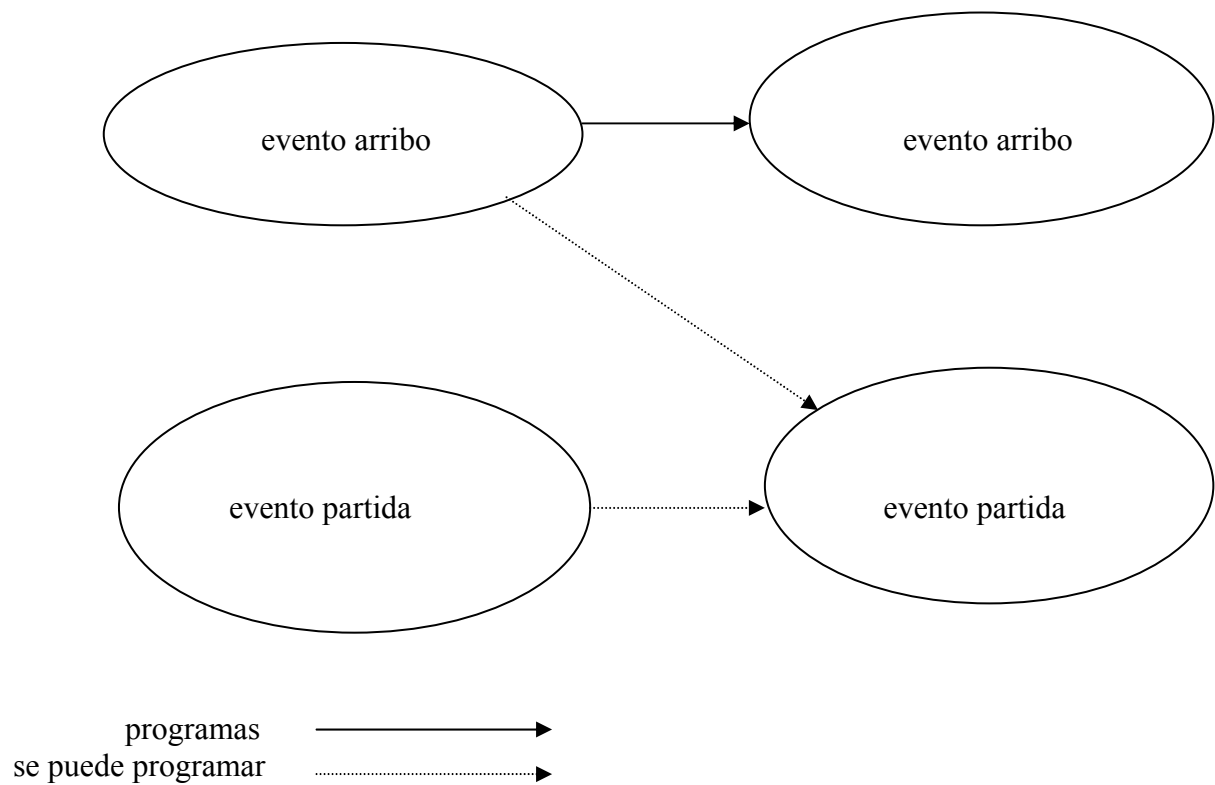
Ejemplo de Panadería Sud

Cuadro de Resultado

	Regla 1	Regla 2	Regla 3
Comprado	7295	7400	7478
Vendido	6500	6835	6835
Utilidades	1426	1568	1548
Promedio simulación de la demanda	36,50		







En el programa del ejemplo un objetivo de la simulación es hallar las estadísticas acerca de tiempo que cada entidad gasta en la cola y en el sistema.  
 Esto no es posible salvo que cada tiempo de arribo de entidad sea registrado.  
 Este tiempo registrado debería ser llevado con la entidad al punto partida de la cola donde puede ser comparado con el tiempo actual de la simulación, TNOW, para proveer el tiempo de espera en la cola.  
 Si sumamos el tiempo de servicio de la entidad al tiempo de cola tenemos el tiempo en el sistema de la entidad.  
 El procedimiento para registrar los tiempos de arribo de las entidades (un atributos de la entidad) utiliza en el programa un arreglo llamado ATRIB.  
 Los ingresos en este arreglo representan las entidades en la cola.  
 En el arribo, el tiempo de arribo de la entidad se coloca al final del arreglo (véase la rutina arribo).  
 Cuando una entidad deja la cola para ir a la estación todas las entidades del arreglo se mueven una posición adelante (véase la rutina partida).  
 Los diagramas clarifican la performance de los programas de computadora.  
 Reflejan los cambios que ocurren en el sistema de cola en cada tiempo de evento.  
 En el diagrama los círculos representan entidades.  
 El número dentro del círculo es el tiempo de arribo de la entidad (atributo de la entidad).  
 CLOCK es el tiempo de simulación (TNOW en los programas).  
 "A" y "D" en la caja representa si el evento es un arribo o una partida, respectivamente.  
 Se asume que el primer arribo es a tiempo 0.  
 Los intervalos entre arribos aleatorios utilizados en los diagramas son:

3,9 2,5 3,6 2,5 1,9 3,3 3,7 3,4

Los tiempos aleatorios de servicio son

1,8 1,5 1,1 4,8 3,8 2,2 3,5 3,7

Clock	Queue	Server	Statistic
0			LQ = 0
Event A	TNA = 3,9 TNE = 1,8	TND = 1,8	NEE = 1 NES = 0 TQT = 0
Clock 1,8			LQ = 0
Event D	TNA = 3,9 TNE = 3,9	TND = 5,4	NEE = 1 NES = 1 TQT = 0
Clock	Queue	Server	Statistic

3,9		3,9	LQ = 0
Event A	TNA = 6,4	TND = 5,4	NEE = 2
	TNE = 5,4		NES = 1
			TQT = 0
Clock 5,4	Queue	Server	Statistic
			LQ = 0
Event D	TNA = 6,4	TND = 7,5	NEE = 3
	TNE = 6,4		NES = 2
			TQT = 0
Clock 6,4	Queue	Server	Statistic
		6,4	LQ = 0
Event A	TNA = 10	TND = 7,5	NEE = 3
	TNE = 7,5		NES = 2
			TQT = 0
Clock 7,5	Queue	Server	Statistic
			LQ = 0
Event D	TNA = 10	TND = 14,8?	NEE = 3
	TNE = 10		NES = 3
			TQT = 0
Clock 10	Queue	Server	Statistic
		10	LQ = 0
Event A	TNA = 12,5	TND = 14,8	NEE = 4
	TNE = 12,5		NES = 3
			TQT = 0
Clock 12,5	Queue	Server	Statistic
	12,5	10	LQ = 1
Event A	TNA = 14,4	TND = 14,8	NEE = 5
	TNE = 14,4		NES = 3
			TQT = 0
Clock 14,4	Queue	Server	Statistic
	14,4 12,5	10	LQ = 2
Event A	TNA = 17,7	TND = 14,8	NEE = 6
	TNE = 14,8		NES = 3
			TQT = 0
Clock 14,8	Queue	Server	Statistic
	14,4	12,5	LQ = 1
Event D	TNA = 17,7	TND = 18,6	NEE = 6
	TNE = 17,7		NES = 4
			TQT = 2,3
Clock 17,7	Queue	Server	Statistic
	17,7 14,4	12,5	LQ = 2
Event A	TNA = 21,4	TND = 18,6	NEE = 7
	TNE = 18,6		NES = 4
			TQT = 2,3
Clock 18,6	Queue	Server	Statistic
	17,7	14,4	LQ = 1
Event D	TNA = 21,4	TND = 20,8	NEE = 7
	TNE = 20,8		NES = 5
			TQT = 6,5
Clock 20,8	Queue	Server	Statistic
		17,7	LQ = 0
Event	TNA = 21,4	TND = 24,3	NEE = 7

D	TNE = 21,4	NES = 6 TQT = 9,6
Clock 21,4	Queue 21,4	Server 17,7
Event A	TNA = 24 TNE = 24,3	TND = 24,3 LQ = 1 NEE = 8 NES = 6 TQT = 9,6
Clock 24,3	Queue	Server 21,4
Event D	TNA = 24,8 TNE = 24,8	TND = 28 LQ = 0 NEE = 8 NES = 7 TQT = 12,5

- TNA    time next arrival    tiempo del próximo arribo
- TNE    time next event    tiempo del próximo evento
- LQ    queue lenght    longitud de la cola
- NEE    number of entities entered    número de entidades que han entrado al sistema
- NES    nunmber of entities served    número de entidades que han recibido servicio
- TQT    total queue time    tiempo total en cola

Nótese que TQT se actualiza sólo cuando deja la cola e ingresa al canal.

Estructura general de los programas de simulación

La estructura del programa simple que vivimos representa la estructura general de los programas de simulación basados en eventos. Los módulos siguientes constituyen un programa de simulación de eventos discretos en un programa desarrollado en un lenguaje orientado a procedimientos (un lenguaje de programación general). Se transcriben en inglés para respetar los conceptos de [Khoshnevis]:

*Main routine* transfers control between the major modules of the program.

*Initialization routine* initializes all variables and clears the statistical data that may have been gathered in a previous run.

*Events timing routine* locates the most imminent future event, advances the simulation clock to the time of the event, and calls the corresponding event-processing routine.

*Future events list (calendar)* contains the list of the unprocessed future events. Other information, such as the attributes of entities causing the event, may also be stored in this structure. Note that in the example program there are a maximum of two future events possible at any point in time. Consequently, there is no need for making an elaborate future events list for this case. If, for example, there were many parallel servers in our queuing system, there would have been a possibility of several upcoming departure events which would have needed to be scheduled (at the start of service times) in the future events list. Since events and their related information constantly appear and disappear in the future events list, a dynamic structure for this list that efficiently utilizes the computer memory is essential for realistic applications. The combination of the event timing routine and the event calendar are often called the *simulation engine*.

*Event processing routines* are individual modules each representing an event in the system. Each module, when called by the events timing routine, makes the proper changes (caused by the event) to the system variables and possibly schedules an event of the same or different type in the future events list.

*Library routines* include a module for pseudorandom number generation and several modules for random variates with various distribution types.

*Statistics routine* collects and processes certain statistics that are specified by the user. These statistics may deal with information concerning the length of queues, the entity waiting time in queues, utilization of facilities, and utilization of resources. The general quantities desired in statistics reports are the mean, standard deviation, minimum, maximum, and last value observed at the end of a simulation. Graphs of frequency histograms and/or plots of variable values over a specified time period may be constructed in the statistics routine.

*Output routine* gathers the values collected by the statistics routine and may perform some operations on these values to create measures such as overall averages. It prints the results and graphs in readable report formats.

Estadística en la simulación

La estadística en los sistemas dinámicos son generalmente de dos tipos: estadística basada en la observación y estadística basada en el tiempo.

La estadística basada en la observación, como implica su nombre, es recolectada en ciertos puntos de observación.

Por ejemplo, el tiempo gastado en el sistema es observado en el momento de partida de la entidad y en ningún otro momento.

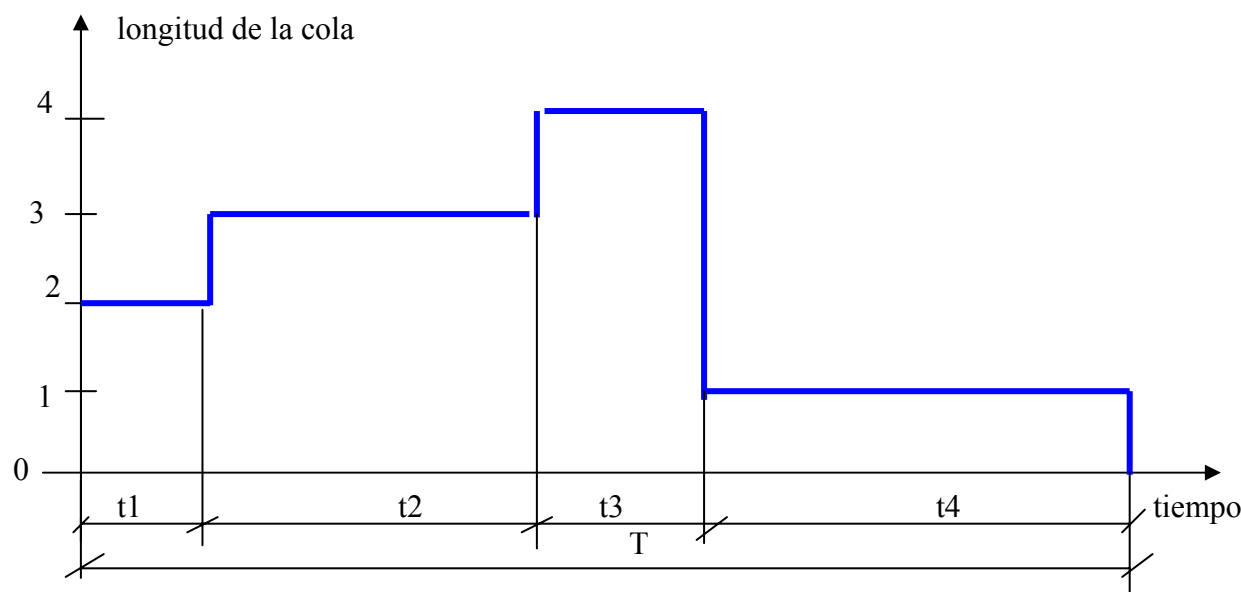
Generalmente, todas las estadísticas que son de dimensión "tiempo" (tiempo de espera en cola, tiempo transversal dos puntos del sistema, tiempo entre partidas, etcétera) están basadas en la observación, y todos los otros tipos de estadísticas son basadas en el tiempo.

Los parámetros básicos, media y varianza, para una estadística basada en la observación (por ejemplo: tiempo de espera) para  $n$  valores observados son:

$$\overline{W} = \frac{1}{n} \sum_{i=1}^n W_i$$

$$Var_w = \frac{1}{n-1} \sum_{i=1}^n (W_i - \overline{W})^2$$

Como ejemplo de estadística basada en la observación considérese:



El tiempo medio de espera en cola es  $12 \cdot 5 / 7 = 1,79$

Esta estadística está basada en los tiempos de espera observados en 7 entidades que completaron el servicio.

Las estadística basada en el tiempo (o promediadas en el tiempo) generalmente se utilizan en cantidades que no representan tiempo.

Por ejemplo, el valor de la variable que representa el número de entidades en una cierta sección del sistema y el número de entidades en cola son representadas como estadísticas basadas en el tiempo.

Un valor confiable de estas estadísticas no puede ser obtenido si se observan sus niveles sólo en ciertas épocas del tiempo.

La aproximación correcta es tomar cada valor tomado por esas variables versus el tiempo que cada valor persiste.

Por ejemplo, mirando la cola en dos puntos en el tiempo, longitudes de cola de 0 y 10 y subsiguientemente concluir que la longitud media de la cola es 5, sería una aseveración incorrecta de la estadística verdadera; la longitud 0 podría haber persistido por sólo un minuto, pero la longitud de 10 por varias horas.

La manera correcta de desconectar estadísticas es verificar longitud de la cola para cada valor que ésta toma y computar el promedio ponderado con el tiempo de esos valores.

El gráfico ilustra un cambio típico en el número de entidades en cola a lo largo de  $T$  unidades de tiempo.

En esta figura  $t_i$  es la longitud de tiempo entre el evento  $(i-1)$  y el  $i$ —ésimo y  $L_i$  es la longitud de la cola al tiempo del  $i$ —ésimo evento.

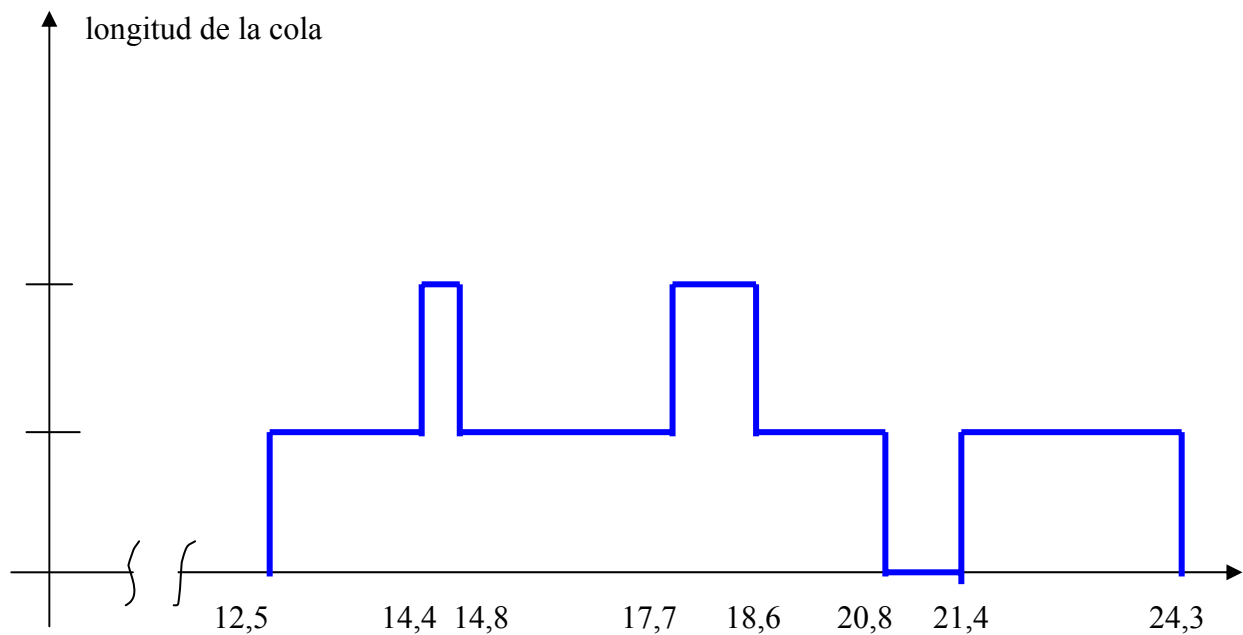
Como muestra la figura cada longitud  $L_i$  de la cola persiste por  $t_i$  unidades de tiempo.

La media y la varianza de la longitud de la cola se encuentran usando las siguientes fórmulas:

$$\bar{L} = \frac{1}{T} \sum_{i=1}^n L_i t_i$$

$$Var_L = \frac{1}{T} \sum_{i=1}^n (L_i - \bar{L})^2 t_i$$

El gráfico siguiente muestra la pauta de cambios en el tamaño de la cola del ejercicio.



En este ejemplo, la longitud promedio de la cola como una estadística basada en el tiempo puede ser calculada de la siguiente manera:

$$\text{Numerador} = (12,5 - 0)0 + (14,4 - 12,5)1 + (14,8 - 14,4)2 + (17,7 - 14,8)1 + (18,6 - 17,7)2 + \\ + (20,8 - 18,6)1 + (21,4 - 20,8)0 + (24,3 - 21,4)1$$

$$\text{Longitud promedio de la cola} = \frac{\text{Numerador}}{24,3} = 0,51 \text{ entidades}$$

En un programa de simulación la suma de las estadísticas y la suma de sus valores al cuadrado pueden ser actualizados cada vez que la variable cambia para computar la media y la varianza de cada estadística al final de la simulación.

Para una variable, digamos X, las relaciones siguientes se utilizan para computar estadísticas basadas en la observación:

$$SUM X_i = SUM X_{i-1} + X_i ; SUM X_0 = 0$$

$$SSQ X_i = SSQ X_{i-1} + X_i^2 ; SSQ X_0 = 0$$

Las ecuaciones equivalentes para estadísticas basadas en el tiempo son:

$$SUM X_i = SUM X_{i-1} + X_i t_i ; SUM X_0 = 0$$

$$SSQ X_i = SSQ X_{i-1} + X_i^2 t_i ; SSQ X_0 = 0$$

Una forma eficiente de recolectar estadísticas basadas en el tiempo es actualizar las estadísticas cada vez que la variable cambia.

Una forma más simple pero menos eficiente es actualizar las estadísticas en cada evento sin importar el cambio en la variable.

Esta aproximación es la utilizada en el programa de ejemplo, donde el valor de la estadística es multiplicada por la cantidad

TNOW — TLE

que es el equivalente a  $t_i$  en las fórmulas de más arriba (TLE es el tiempo del último evento).

En ambos casos las estadísticas son ponderadas por los segmentos de tiempo entre las observaciones pasada y presente.



El total acumulado de todos los valores ponderados entonces es dividido por el lapso sobre el cual se cosechan las estadísticas.

La utilización del canal se oculta mejor utilizando una variable binaria que toma los valores 0 y 1 para las condiciones de desocupado y ocupado.

La utilización promedio es como máximo 1 en un sistema con un canal.

Para un sistema con múltiples servidores en paralelo la utilización media puede ser tan alta como el número de canales el sistema.

Para la compañía de transporte (nuestro ejemplo) la utilización promedio en los diez días de simulación bajo el primer escenario es 1.

Para el segundo guión de este ejemplo tenemos:

$$\text{Utilización promedio del primer lugar} = \frac{2,5 + 3 + 2}{10} = 0,75$$

$$\text{Utilización promedio del segundo lugar} = \frac{2 + 1,5 + 1}{10} = 0,45$$

Canales que se espera que estén en uso en cualquiera de los lugares

$$0,75 + 0,45 = 1,2 \text{ canales}$$

Las estadísticas de "tiempo ocupado" de los canales son estadísticas basadas en la observación de la duración de estado ocupado ininterrumpido.

Cada estadística conservada es el segmento de tiempo que comienza en la transición desocupado—ocupado y termina en la siguiente transición ocupado—desocupado (en el cual el punto de dato se observa).

Las estadísticas de "tiempo desocupado" se recolectan de la misma forma.

Considérese, por ejemplo, el segundo guión del ejemplo de la compañía transportista.

El tiempo de ocupado promedio de cada canal se calcula como sigue:

$$\text{Tiempo promedio ocupado canal uno} = \frac{2,5 + 5}{2} = 3,25 \text{ días}$$

$$\text{Tiempo promedio ocupado canal dos} = \frac{2 + 1,5 + 1}{3} = 1,5 \text{ día}$$

$$\text{Tiempo promedio ocupado del sistema} = \frac{3,25 + 1,5}{2} = 2,375 \text{ días}$$

Los tiempos promedio de desocupado se hallan de la misma forma:

$$\text{Tiempo promedio desocupado canal uno} = \frac{1,5 + 1}{2} = 1,25 \text{ día}$$

$$\text{Tiempo promedio desocupado canal dos} = \frac{2 + 1 + 1,5 + 1}{4} = 1,35 \text{ día}$$

$$\text{Tiempo promedio desocupado del sistema} = \frac{1,25 + 1,35}{2} = 1,3 \text{ día}$$

## Software para simulación

En estos momentos las herramientas para simulación de eventos discretos se clasifican en tres categorías.

El autor utiliza EZSIM, una nueva herramienta de simulación de propósito general.

## EZSIM - Una herramienta de simulación de sistemas discretos de propósito general

Estas tecnologías buscan llevar la simulación directamente a usuarios finales con diferentes habilidades en computación.

Los analistas de simulación se benefician con la mayor velocidad de construcción de modelos y de verificación que ofrecen estos entornos.

Los modelos en EZSIM se representan en forma de red.

Cada nodo en la red del modelo representa un proceso y los arcos muestran el camino de la entidad de un nodo a otro.

Trabaja en dos modos

- stand—alone y
- generador de programas, que genera códigos de alto nivel para varios lenguajes populares de simulación.

## Menú de apertura de EZSIM

EZSIM comienza con un menú de apertura que permite crearlos, recuperados, listarlos, cambiarles el nombre y borrarlos.

Se puede llamar archivos de diferentes drives o directorios especificando el código de letra del drive y el camino del directorio (por ejemplo A:\EZ\MYFILE).

Después del nombre del archivo no se admite extensión EZ.

Después de que el usuario especifica el nombre del archivo del sistema que será modelado se invoca la pantalla gráfica de EZSIM.

Esto provee un entorno para la creación o modificación de la red.

Todos los archivos se guardan con extensión EZ.

## Construcción de la red

Se usa la pantalla gráfica.

Primero se pide al usuario que identifique el encabezado opcional del proyecto.

Presionando H cuando se está en el modo gráfico muestra la pantalla de ayuda.

Se pueden ver en esta pantalla todos los parámetros de control gráfico y las definiciones de modo.

Para ver la definición de cada uno de los nodos se debe presionar la mayúscula resaltada en el nombre del nodo.

Por ejemplo, presionando S proveerá la definición del nodo SOURCE.

Alternativamente se puede ver una ventana de definición del nodo llevando a él el cursor y presionando ENTER.

Si se presiona ESC en el modo ayuda, vuelve la pantalla al modo gráfico.

Mientras se está en modo gráfico, si se presiona la tecla del código del nodo hace que el cursor vaya al nodo correspondiente.

La abreviación nodal puede verse en una columna a la derecha presionando la tecla F1.

Después de ubicar cada nodo en la red, se pide el nombre del nodo.

Los nodos se pueden conectar ubicando el cursor (por teclas de movimiento o por ratón) en el nodo y presionando la tecla —.

Este nodo entonces se conecta automáticamente con el nodo siguiente en el cual se presione —.

Se hace un chequeo en línea para asegurarse que las conexiones están permitidas.

EZSIM analiza cada nodo y sus posibles relaciones con otros nodos de la red.

También detecta errores lógicos en los pasos iniciales de la construcción del modelo.

Por ejemplo, conectando un nodo QUEUE a un nodo TERMINATE genera un incitador de EZSIM indicando que esa conexión carece de sentido.

Nótese que cuando el usuario especifica los nodos origen y destino para su conexión, EZSIM determina automáticamente el camino de conexión y hace la conexión.

Este proceso automático acelera el proceso de construcción de la red.

En modelos complejos, un área de la red puede volverse confusa, con varios nodos y líneas de conexión, donde los inputs y outputs hacia y desde los nodos pueden no ser identificables.

En este caso, uno puede específicamente elegir ver los inputs y los outputs de un nodo seleccionado.

Los arcos deseados se pueden ver colocando el cursor en el nodo y presionando ALT+I, que hace parpadear todos los arcos de input al nodo en un color diferente.

Presionando ALT+O hace parpadear los arcos salientes.

Presionando la barra espaciadora se detiene la acción de parpadeo.

La red EZSIM puede ser tan grande como permita la computadora (su memoria).

Si la red es demasiado grande para entrar en la pantalla se puede hacer correr arriba, abajo, a la izquierda y a la derecha.

El indicador de navegación de la red en la parte superior izquierda indica la porción de la red que no es mostrada en pantalla.

Las flechas parpadeantes en la esquina superior izquierda indican la posición de estas porciones de la red respecto al área de pantalla visible.

Esta capacidad es especialmente útil cuando se están haciendo redes disjuntas, dado que de otra manera uno puede perder alguna de esas partes disjuntas cuando está viendo o construyendo un modelo grande.

## Declaración y especificación de parámetros nodales.

Hay ciertas declaraciones y parámetros que deben ser especificados para cada nodo de la red.

**El usuario puede elegir un nodo en la red con el cursor en el nodo y presionando ENTER.**

Esto invoca una ventana para la identificación paramétrica y declarativa del nodo.

Se generan una serie de preguntas relevantes (acompañadas con listas de respuestas válidas e incitadores de ayuda sensibles al contexto) y menús de elecciones en cada nivel de pregunta relacionada con la identificación de los parámetros de cada nodo elegido.

Por ejemplo, algunas distribuciones de probabilidad estándar se proveen en un menú que aparece automáticamente cada vez que se lo necesita.

Los nodos se pueden elegir en cualquier secuencia.

La información ingresada para un nodo puede afectar las condiciones de otros nodos relacionados.

En tal caso EZSIM muestra la relación al usuario y pide aclaración.

Cada nodo colocado en la pantalla es identificado por un código y el nombre dado por el usuario en un rectángulo de una sola línea.

Un signo de interrogación aparece en la parte inferior izquierda de la pantalla tan pronto como se ubica el primer nodo en la pantalla.

Esto indica que hay algunos nodos para los cuales la información paramétrica no ha sido aún especificada.

Esto es útil cuando la red es mayor a la pantalla dado que señala la existencia de rectángulos de línea sencilla en partes invisibles de la red.

La identificación de información paramétrica y declarativa (variables y atributos) sin embargo puede no ser posible si el proceso de identificación para otro nodo relacionado es incompleto.

Esta es una consideración deliberada del diseño de EZSIM para ayudar al usuario a que no evite algunas declaraciones importantes necesarias para completar el modelo.

EZSIM señala tales situaciones y ayuda al usuario a crear un modelo completo.

**Cualquier identificación global (su proceso) puede ser diferida temporariamente para más tarde si se presiona ESC durante el proceso.**

Este esquema permite al usuario concentrarse primero en la arquitectura global del modelo y luego elaborar los detalles.

Después de la selección del modelo, se muestra una ventana dedicada que guía al usuario a identificar cada uno de los parámetros del nodo en cuestión.

Para cada ítem de la lista, generalmente hay una ventana de ayuda que se muestra en la caja inferior de la pantalla.

Presionando la tecla RETURN después de ingresar cada parte de información lleva el cursor a la pregunta siguiente.

En ciertos casos, la naturaleza de la pregunta hecha en cada estadio sucesivo depende de la elección de respuesta a la pregunta actual.

En la mayoría de las instancias las elecciones posibles de respuesta puede ser seleccionada utilizando una barra corriente y presionando ENTER; en muchas instancias se da una elección por defecto para la respuesta entre < >.

En esos casos presionando RETURN sin ingresar resultado alguno resulta en la elección del valor por defecto.

Nótese que las teclas retroceso o delete se pueden usar para cambiar respuestas dadas previamente a las preguntas de identificación de parámetros.

Después de completar las respuestas a todas las preguntas relativas al nodo, se devuelve el control a la pantalla gráfica para la selección del próximo nodo o para continuar la edición y construcción de la red. Si se trata de salir del modo red (presionando ESC) mientras la identificación de parámetros o el proceso de construcción de la red es incompleto, surge una ventana en la esquina inferior derecha que advierte al usuario de la situación.

Esta ventana da la elección de o retornar a la red, salvando el modelo para ser completado en algún punto futuro del tiempo o abandonar el modelo sin guardarlo.

Presionando la tecla ESC después de completar la etapa de identificación de los parámetros nodales da comienzo a la etapa de inicialización del sistema.

Esta etapa de inicialización ocurre sólo si hay cantidades (tales como variables del usuario y recursos) que necesitan ser especificadas.

**Debe remarcarse que el proceso de construcción del modelo puede ser abortado en cualquier momento presionando CTRL+C que hace que se regrese al menú de entrada.**

## Especificación de estadísticas

Después de especificar los parámetros del nodo, se despliega la pantalla de estadísticas.

Aquí el usuario puede especificar las estadísticas deseadas por medio de cualquier especificador que aparecerá en el output de simulación.

Un menú de selección permite al usuario elegir las estadísticas deseadas de la siguiente lista de opciones:

- Tiempo de arribo de la primera entidad al nodo.
- Todos los tiempos de arribos de entidades al nodo.
- Tiempo de travesía entre dos nodos.
- Tiempo entre arribos.
- Conteo de entidades en el nodo.
- Cualquier variable definida por el usuario.
- Una expresión.

Una vez que se ha hecho la selección, se le pregunta al usuario dónde se aplica la selección en el sistema.

Por ejemplo, si la opción elegida es el tiempo para hacer la travesía, se le pregunta al usuario para que identifique en el sistema cuáles son los nodos entre los cuales se desean las estadísticas del tiempo de travesía.

En este caso aparece en una ventana un listado de todos los nodos de entre los cuales el usuario puede elegir dos.

Para algunas estadísticas, se requiere la identificación de si las estadísticas debe ser consideradas basadas en el tiempo o basadas en la observación.

Para un modelo pueden ser utilizadas varias páginas de menús de estadísticas.

Presionando ESC mientras se está en la pantalla de estadísticas llevará al menú de control.

Debe mencionarse que EZSIM recolecta automáticamente las estadísticas de todo tipo de colas, instalaciones y recursos especificados en el modelo.

Las estadísticas en el output se identifican bajo el nodo asociado o nombres de recursos dados en el estadio de modelizado.

## Menú de control/ejecución del modelo EZSIM

Después de que se hayan completado los estadios mencionados más arriba, se presenta el menú de control de EZSIM.

El menú de control es el puente a todos los estadios precedentes de construcción del modelo y a los estadios de ejecución y output.

El retorno a la red del modelo o al menú de estadísticas se puede hacer por medio de una tecla.

Como se dijo, el modelo puede ser ejecutado al modo batch o al modo animado.

Si se elige modo batch el menú de control, se debe identificar la longitud deseada de la simulación en unidades de tiempo.

La longitud del período transitorio después del cual las estadísticas deseadas debe ser recolectadas deberían ser dadas también a EZSIM.

Entonces comienza el proceso de simulación y se muestra, mientras progresa, el tiempo corriente de simulación.

El usuario puede detener momentáneamente la simulación en cualquier momento y observar los outputs durante la ejecución del proceso.

Se puede retomar la ejecución después de observado el output.

Si el modo de ejecución seleccionado es animación, el usuario debe identificar las entidades animadas por medio de su elección de caracteres y de su color.

La animación puede ser observada o en el modo continuo o en el modo a pasos.

Entonces el control regresa a la red y el entonces se muestra el movimiento de las entidades en los arcos.

Las redes que son más grandes que la pantalla se pueden mover en la dirección deseada.

Mientras progresa la animación, el tiempo real de la simulación se muestra en la esquina superior derecha de la pantalla.

La velocidad del movimiento de los símbolos de entidades en el modo animado puede ser cambiado tecleando una vez.

La capacidad de animación de la red de EZSIM provee una forma simple de verificación del modelo.

Durante el curso de la animación pueden verse el movimiento de las entidades mientras recorren arcos y robos de acuerdo con la lógica del modelo, el estatus del modelo, y los valores cambiantes de los niveles de recursos y de las variables del usuario.

El proceso de animación puede ser interrumpido en cualquier lugar para revisar el output de la simulación.

El usuario puede seguir el proceso de animación después de observar el output deseado.

## Menú output de EZSIM

El menú de output permite la observación de las estadísticas seleccionadas y gráficos en pantalla durante o al final tanto del modo batch como del modo de ejecución con animación.

El output de simulación y la red del modelo con la información relacionada puede ser guardada en disco o enviada a impresora por medio de la selección apropiada del menú output.

El recomienzo tanto de la ejecución batch (si fue interrumpida antes del fin del tiempo estipulado de simulación) como de la ejecución animada puede ser hecho mediante el menú de output.

El tiempo batch de ejecución también puede ser extendido por el lapso deseado seleccionando la opción asociada en el menú de output (nótese que la extensión del tiempo de simulación se aplica sólo al modo de ejecución batch).

El menú output también tiene un puente al menú control.

## Modificación del modelo

Después de correr el modelo EZSIM, el usuario puede volver a cualquier sección del entorno a través del menú de control.

Una de las mayores capacidades de EZSIM es su facilidad de modificación de la red del modelo y la información relacionada al modelo.

Las modificaciones pueden ser hechas en cualquier punto del proceso de construcción del modelo o después de terminado.

En tales casos, EZSIM detecta todos los componentes que puede ser afectados por esos cambios; si se necesita información adicional para mantener la integridad del sistema, se consulta al usuario.

Se pueden agregar o eliminar los y se puede modificar como se desee la información paramétrica o declarativa.

Cada vez que se efectúa un cambio en la estructura de la red, los nodos afectados cambian a un rectángulo de línea simple, significando esto que la información paramétrica y/o declarativa relativa a los grupos afectados por los cambios necesitan ser modificados.

Por ejemplo, si a la red se le añade un nodo y uno de los nodos existentes se conecta al nuevo nodo, el nodo existente cambia a rectángulo de una sola línea y aparecerá un ? en la parte inferior derecha de la pantalla.

La nueva información de relación para el nodo afectado por el cambio será subsecuentemente identificada.

Un análisis realista de simulación siempre requiere varias modificaciones al modelo original para testear varias configuración está de sistemas y sus performances asociadas.

Este proceso puede ser complejo cuando el método usual de simulación.

EZSIM hace esta tarea tan simple como sea posible y provee medios para el monitoreo automático de la estructura del modelo.

EZSIM trata de prevenir al usuario de hacer errores lógicos o estructurales así como errores sintácticos a través de la construcción del modelo y los procesos de modificación.

## Demostración del procedimiento de aplicación EZSIM

El nodo **SOURCE** crea entidades.

Cada entidad que crea el nodo **SOURCE** puede tener un nombre, el cual se torna un atributo para la entidad (llamado **NAME**).

Si no se especifica un nombre para la entidad se la mostrará como un ? en la simulación.

Es posible especificar el número total de entidades a ser creadas por el nodo **SOURCE**.

El nodo **DELAY** se utiliza para crear una demora que corresponde al tiempo de travesía de una entidad de un nodo a otro.

El tiempo de travesía se selecciona de la ventana estándar de tiempo (se describe más abajo) y puede ser representada por varias distribuciones estadísticas, un valor de atributo, un valor de una variable del usuario o una expresión.

El nodo **TERMINATE** finaliza el camino de las entidades.

Cada nodo **TERMINATE** puede terminar la simulación completa si el contador de entidades especificado en él es alcanzado.

El nodo **FACILITY** actúa como un servidor.

Las entidades permanecen en el nodo durante la duración de su servicio.

Cuando el nodo está ocupado, las entidades que arriban deben esperar hasta que el nodo se libere.

El nodo **FACILITY** debe ser precedido por un nodo **QUEUE**.

A la entidad que deja el canal se le puede dar un nombre nuevo.

Esto posibilitará, por ejemplo, mostrar las partes procesadas y no procesadas con símbolos diferentes en el modo animación.

Se pueden especificar múltiples canales en paralelo para una instalación dada.

El nodo **QUEUE** representa buffers antes de los nodos **FACILITY** y siempre es sucedido por un nodo **FACILITY**.

Un **QUEUE** puede tener limitación de capacidad, varias disciplinas de prioridad tales como primero—entra—primero—sale (FIFO), último—entra—primero—sale (LIFO), y así otras.

A través de EZSIM, varios elementos de tiempo como tiempo de la creación de la primera entidad, tiempo entre creaciones y tiempos de demora en los nodos **DELAY** y **FACILITY** tiene una ventana estándar que permite su especificación.

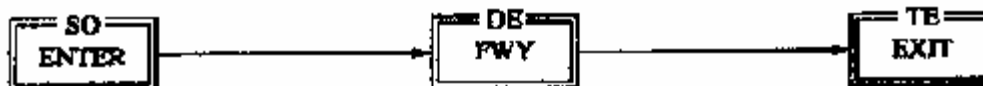
Esta ventana ofrece varias opciones.

Cuando se especifica la opción de distribución estadística, una ventana que enlista varios nombres de distribuciones permite seleccionar la distribución requerida.



Después que se elige la distribución, el usuario es incitado con mensajes preguntándole los valores de los parámetros de la distribución y la elección de la semilla del número al azar (por el momento elijase la semilla por defecto).

### Ejemplo 3.1



**FIGURE 3.16**  
EZSIM network model of the freeway example.

Supóngase que los autos arriban a la entrada de un segmento de una autopista con un tiempo entre arribos distribuido exponencialmente con una media de 3 minutos.

El tiempo de viaje a través del segmento de autopista se distribuye normalmente con una media de 15 min y un desvío estándar de 2 min.

Estamos interesados en observar, utilizando animación, los cambios en el número total de autos en la autopista durante un lapso seleccionado.

La red de este modelo se muestra en la Fig. 3.16.

Para crear este modelo inicie EZSIM, seleccione la opción Build/Retrieve en el menú de entrada y dele un nombre, digamos EX1, a su modelo.

En la pantalla gráfica presione S para poner un nodo SOURCE en la posición del cursor.

Se le pedirá un nombre para el nodo.

El número máximo de caracteres para el nombre de nodo es 6.

El nombre dado, digamos ENTER, aparecerá dentro del nodo.

Mueva el cursor a la derecha, a distancia suficiente del nodo SOURCE, y presione D para colocar un nodo DELAY en la posición del cursor.

De la misma manera coloque un nodo TERMINATE en el lugar apropiado.

Ahora ubique el cursor en el nodo SOURCE y presione -.

Ubique el cursor en el nodo DELAY y presione de nuevo -.

Los dos nodos serán conectados con una línea.

De la misma manera conecte el nodo DELAY al nodo TERMINATE.

Presione H para obtener ayuda en cualquier estadio de la construcción para ver los códigos de operación que pueda necesitar (deshacer la última acción, mover un nodo, etc.).

Para completar su modelo de simulación debe identificar la información paramétrica para cada nodo.

Puede seleccionar los nodos en cualquier secuencia ubicando el cursor sobre él y presionando la tecla ENTER para observar la ventana asociada de parámetros.

Puede comenzar con el nodo SOURCE en este ejemplo.

Cuando seleccione este nodo aparece una ventana.

La primera pregunta que se hace en esta ventana es el nombre de la entidad.

Puede darle el nombre CAR a las entidades generadas en el nodo.

Fije el primer tiempo de creación en cero presionando cero y luego Enter en el incitador de tiempo de la ventana.

Para identificar el tiempo entre creaciones seleccione la opción Statistical Distribution en la ventana de tiempo y seleccione la distribución exponencial.

Ingrese 3 como media y seleccione la semilla de número por defecto.

Presione Enter en todos los incitadores para seleccionar los valores por defecto.

De la misma manera, seleccione la distribución normal para el tiempo en el nodo DELAY, e ingrese los parámetros 15 y 2.

Elija la opción por defecto para el conteo de terminación en el nodo TERMINATE.

En este punto desaparece el signo de interrogación en la esquina inferior derecha, lo que significa que el modelo está completo.

El presionar ESC le lleva al Statistics menu.

Para ver la función de este menú, elija un nombre de estadística, digamos TRIP.

Esta estadística es para denotar el viaje del auto desde los puntos de entrada hasta el de salida de la autopista.

La elección de estadística en el menú que salta es, por lo tanto, "Traversal time between two nodes."

EZSIM asigna automáticamente las estadísticas del tipo basado en la observación a su elección, y le pide que indique los dos nombres de nodo entre los cuales desea las estadísticas del tiempo de travesía.

Después de seleccionar los dos nodos seleccione No default para la pregunta relativa a gráfico, y presione ESC para ir al menú de control.



En el menú de control elija la opción Animate, la cual le incita a seleccionar un símbolo para la entidad llamada CAR.

Puede elegir cualquier carácter presionando la tecla relacionada.

Entonces se le dan las opciones para los colores de frente y fondo de pantalla.

Simplemente mueva la flecha a la elección deseada y presione ENTER.

Ahora puede correr su modelo en animación continua presionando C.

Observe el modelo mientras corre.

El número de entidades en el nodo DELAY se muestran en él.

Este número cambia cuando algunos autos arriban y otros parten.

Mire el tiempo de simulación en la esquina superior derecha de la pantalla mientras avanza de un tiempo de evento a los próximos.

Puede acelerar la animación o disminuirla usando las teclas PgUp y PgDn.

Presionando las teclas de flechas se mueve la red entera y las entidades animadas.

Si presiona ESC durante la animación verá el menú Output.

Seleccione la opción Variables en este menú para ver las estadísticas llamadas TRIP, las cuales ha especificado antes.

Para regresar a la animación seleccione la opción Resume del menú Output.

Seleccione la opción Control menú del menú Output, y vaya de allí a gráficos para modificar su modelo.

Seleccione el nodo SOURCE y modifique el número de creaciones de entidades de infinito a un número limitado, digamos veinte.

Anime el nuevo modelo y observe su operación.

Experimente dándole un número limitado también para el conteo de terminación.

Después de completar este ejemplo, estará preparado para utilizar EZSIM para experimentar con modelos más grandes que pueden involucrar varios tipos de procesos.

### Ejemplo 3.2

En este ejemplo se estudia un sistema de cola similar a ejemplo que vimos antes.

Suponga que los clientes arriban a un canal para recibir servicio.

El tiempo entre arribos de clientes se distribuye uniformemente entre 10 y 20 mim.

El tiempo de servicio al cliente se distribuye uniformemente entre 8 y 15 mim.

Suponga que el buffer anterior al canal puede acomodar prácticamente un número ilimitado de personas.

Supongamos que estamos interesados en simular el sistema para 500 entidades que arriban o 10.000 minutos de operación, cualquiera que ocurra antes, para encontrar las estadísticas de la cola y el canal.

La Figura 3.17 muestra el diagrama de la red de los sistemas.

Presionando las teclas Q y F coloca a los nodos QUEUE y FACILITY, respectivamente, en la pantalla gráfica.

Los parámetros para los nodos SOURCE y TERMINATE se seleccionan de la misma manera del ejemplo anterior.

En el nodo SOURCE el número máximo de creaciones se fija en 500.

Se puede dar el nombre CUST a la entidad en el nodo SOURCE.



**FIGURE 3.17**  
Demonstration of queue and facility nodes.

En el estadio de especificación de parámetros del nodo QUEUE, se deberían seleccionar los valores por defecto para todos los campos en la ventana relacionada.

Para el nodo FACILITY, se debe identificar un canal con la duración de servicio especificada.

Se deberían seleccionar valores por defecto para otras opciones.

Se le podría dar un nuevo nombre, digamos SCUST (para representar serviced customer), a la entidad que deja el nodo FACILITY.

Para hacerlo, se debería responder con Y a la pregunta "Would you like to change the entity name?" en la ventana FACILITY.

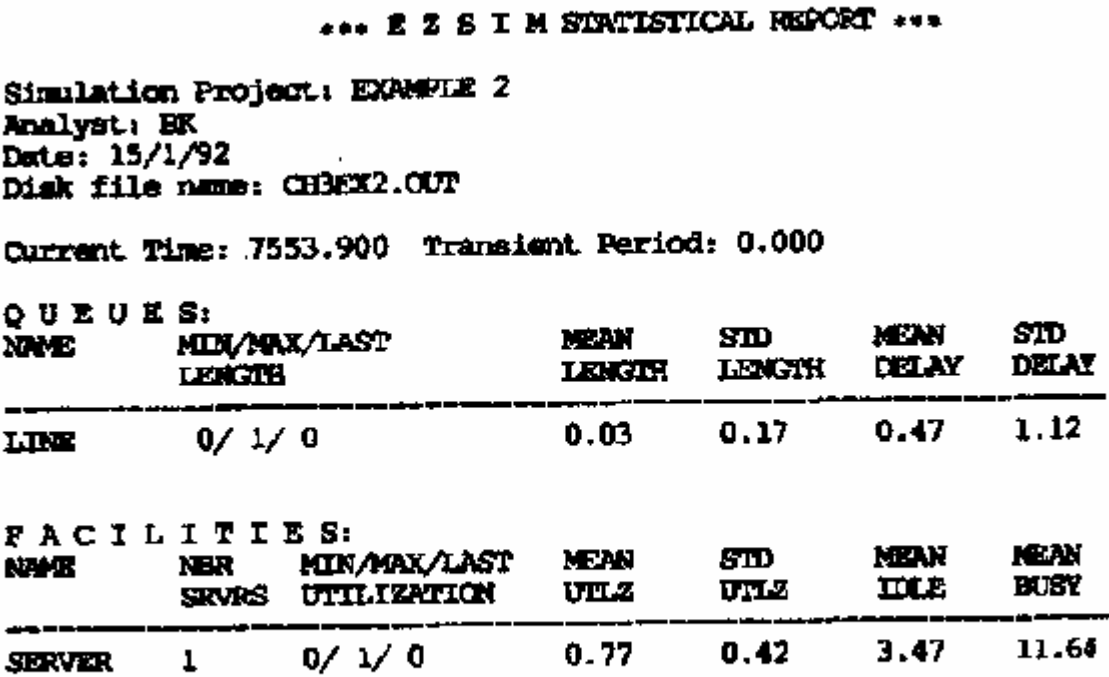
Entonces puede ser ingresado el nombre SCUST.

Después del estadio de construcción de la red, aparece el menú Statistics.

Dado que en este ejemplo estamos interesados sólo en estadísticas de la cola y del canal, que son recolectadas e informadas automáticamente por EZSIM, se puede salir del menú Statistics presionando ESC.

En el menú Output el modelo se puede observar en modo animación.

Especifique los símbolos de entidad C y S para clientes atendidos y no atendidos, respectivamente. Note que en el curso de la animación los números en la parte inferior de los nodos QUEUE y FACILITY representan el número corriente de entidades en el nodo correspondiente. El modelo puede ser también corrido en modo batch. La longitud de la simulación de 10.000 min puede ser especificada después de seleccionar la opción "Run in batch mode" del menú Control. Se puede especificar un período transitorio después del cual EZSIM comienza a recolectar las estadísticas. En este estadio se puede seleccionar un valor por defecto de cero para el período transitorio. La Figura 3.18 muestra el output de la simulación para este modelo. Nótese que en este caso la simulación ha concluido antes del tiempo límite de 10.000 min. Esto es así porque las 500 entidades ingresaron al sistema antes de este tiempo. El output refleja varias estadísticas como mínimo, máximo, y el último número de entidades observadas en la cola, valores medio y desvío estándar (STD) para la longitud de la cola, tiempo de espera en cola y utilización del canal, y promedio de los tiempos de ininterrumpidamente ocupado y desocupado del canal. Para crear este output, se elige la opción Print Output en el menú Output después de que se haya completado la ejecución. Seleccionando la opción Print Model en el menú Output resulta en la impresión de la red y la lista de información relativa a cada nodo en el modelo.



**FIGURE 3.18**  
Output of the queuing system simulation.

**El proceso de simulación**

Los estadios siguientes pueden ser identificados como los principales que un analista debe seguir para obtener un estudio completo y exitoso de simulación.

- *System realization.* Definition of the purpose of the study and subsequent abstraction of the reality leading to system identification (identification of the system boundary and components and the nature of the relationships among the components). This abstraction process is largely an art.
- *Data acquisition.* Identification, collection, and proper representation of the data, describing outside influences and behavior of some internal processes.
- *Model construction.* The abstraction of the system into a representative models that to the best extent possible and desirable mimics the system behavior. Modeling tools such as simulation programming environments may be used at this stage; therefore, the simulation software selection task may be included in the model construction phase.
- *Verification.* The process of establishing that the computer implementation of the model is error-free.

- *Validation.* The process of establishing that the model and the data correctly represent the important aspects of the system (an error-free computer program does not always represent a valid model).
- *Experimentation.* The process of devising relevant and efficient experimental conditions under which the model behavior is examined. This stage may require parametric changes as well as structural changes to the model; that is, model reconstruction may be performed at this stage.
- *Analysis.* The process of making sense out of the simulation output data for each of the experimental scenarios, and comparison of scenarios.
- *Documentation.* The process of describing the problem and the methodology used to address it with minimal use of technical simulation jargon, and the translation and summary of the results of the simulation output analysis into a useful information format that relieves the end user of the task of digging the facts out of computer printouts. The documentation process may include recommendations for the implementation stage.
- *Implementation.* The process of making decisions that lead to changes in an existing system or to the construction of a new system on the basis of the simulation study. This stage may or may not involve the simulation analyst.



## Una revisión de procesos comunes en sistemas discretos

Las herramientas de simulación orientadas a procesos tratan de proveer módulos de programación pre—escritos y pre—compilados que representan fenómenos (procesos) de ocurrencia común con varios guiones realistas, aliviando así a los constructores de modelos de la tediosa tarea de reconstruir frecuentemente estos segmentos de programas para ser usados en sus proyectos de simulación.

La teoría de programación de la simulación se reduce a seleccionar apropiadamente y unir lógicamente esos módulos listos para usar y especificar los valores paramétricos relacionados.

Lo que puede ser visto como un proceso común es muy subjetivo y específico para el dominio de aplicación.

O sea, no todos los lenguajes de simulación orientados a procesos tienen el mismo conjunto de módulos como un repertorio único de procesos comunes.

De cualquier manera, hay una gran superposición en las ofertas de todos esos lenguajes que incluyen cierto número de procesos casi similares.

La base para esta simulación es el concepto de movimiento/acumulación del bien físico, que es universal y se aplica a todos los dominios de aplicación en los cuales está presente el movimiento de entidades.

Ciertas ocurrencias suceden frecuentemente en los dominios en los cuales se usa a veces el análisis de simulación (manufactura, operación de oficinas, flujo de tránsito, transporte, gerenciamiento de construcción, etcétera).

Por lo tanto, varios lenguajes tratan de incluir algunas capacidades que se aplican a lo específico de estos dominios de aplicación.

Los lenguajes de simulación orientados a procesos pueden diferir en su estructura sintáctica (tales como los nombres utilizados para representar cada proceso común, parámetros asumidos y valores por defecto y el formato de las sentencias de sus programas), y en las capacidades de los módulos que representan los procesos del mundo real considerados en el lenguaje.

### Movimiento de Entidades

El dinamismo en los sistemas discretos es causado por el movimiento de entidades que resulta en la ocurrencia de eventos que cambian el estado del sistema a lo largo del tiempo.

En la mayoría de los sistemas las entidades entran al sistema como inputs a través de los bordes del sistema, se mueven entre los componentes del sistema y quizás dejan los límites del sistema en forma de outputs del sistema.

En algunos casos el sistema puede contener inicialmente algunas entidades que se mueven a través del sistema, circulan entre los componentes del sistema y quizás dejan (totalmente o en parte) los bordes del sistema.

Alternativamente, algunas entidades pueden que nunca dejen el sistema (ejemplo: pallets en el almacén, equipo de movimiento y construcción en una obra)

### Creación de entidades

Para crear entidades como ingresos al sistema (como clientes ingresando a un banco o autos llegando a una intersección) se debería considerar un proceso común que actúe como fuente de entidades.

El módulo que representa este proceso es llamado fuente, crear, generar y cosas por el estilo en los diferentes lenguajes de simulación.

Las propiedades deseables de este módulo son:

El módulo debería permitir fijar el *tiempo de creación de la primera entidad*. Este valor puede ser una constante o una variable aleatoria.

El módulo debería proveer el modo de seleccionar el *tiempo entre arribos*, que puede ser una constante o una variable aleatoria. El módulo debería proveer un generador de variables aleatorias que provea muestras de varias funciones de distribución de probabilidad teóricas. El usuario debería poder seleccionar fácilmente la elección indicando sólo el nombre de la distribución y los valores de sus parámetros. También deberían hacerse previsiones para distribuciones de probabilidad a medida del usuario.

El módulo debería permitir *asignar un nombre* a las entidades creadas. Este nombre será asignado a la entidad como un atributo. Esta información puede ser usada en segmentos posteriores en la determinación de la ruta de la entidad a través de varias ramas en base al tipo de entidad y para facilitar el tratamiento especializado de entidades diferentes que entran al mismo segmento del sistema.

Sería deseable que el módulo permitiera *crear entidades en el modo batch (batch creation)*, o sea, la creación de más de una entidad al final de cada lapso intercreación.

El módulo debería proveer el *detener la creación de entidades después de que un cierto número* que haya sido creado. Entre otras ventajas, esta capacidad provee control de la longitud del proceso de simulación por medio del número de entidades más que por tiempo.

El módulo debería permitir el detener la creación de entidades *después de una cierta cantidad de tiempo* haya pasado.

Como se ha mencionado antes, las entidades pueden estar inicialmente en el sistema en lugar de ingresar desde fuera.

En tales casos, generalmente las entidades se ubican inicialmente en los puntos de acumulación.

Los módulos que corresponden a los puntos de acumulación (colas) deberían proveer la forma de inicializar el número deseado de entidades en cada uno de esos sistemas componentes.

Deberían también permitir fijar los atributos de tales entidades a medida que el usuario las inicializa en el sistema.

### **Terminación de entidades**

En la mayoría de los sistemas, las entidades que ingresan al sistema también lo dejan.

El módulo que representa esta partida o desaparición de entidades es llamado terminar, pozo, partida, fuera y así siguiendo por los diferentes lenguajes de simulación.

Una propiedad deseable del módulo que representa el proceso de terminación es la habilidad de especificar el número de terminaciones de entidades requeridas para terminar el proceso de simulación.

La posibilidad de controlar la creación de entidades en el módulo fuente por número o por tiempo, y el número de terminación de entidades que se necesitan en el módulo de terminación para finalizar la simulación, provee al analista la opción de terminar la simulación cuando el sistema está en los estados tanto de vacío como de no vacío.

Por ejemplo, si un modelo particular de simulación utiliza un módulo fuente y uno de terminación, entonces por medio de la especificación de un cierto número máximo de creaciones en la fuente, digamos  $N$  entidades, y no especificando un límite de conteo de terminación en el punto de terminación, el módulo fuente deja de crear entidades cuando se hayan creado  $N$  entidades, pero la simulación continúa hasta que todas las entidades hayan dejado el sistema a través del módulo de terminación.

En este caso la simulación termina cuando el sistema está en el estado vacío.

De manera similar, si el módulo fuente debe terminar la creación de entidades en cierto momento, y no se especifica un número en el proceso de terminación, entonces la simulación continúa hasta que todas las entidades dejan el sistema.

Esta característica es útil en la simulación de escenarios tal como el de un banco que cierra sus puertas en un cierto momento a los clientes que arriban pero continua atendiendo a los que ya están dentro.

Luego la operación del banco termina cuando hayan sido atendidos todos los clientes.

Alternativamente, si el número de entidades especificadas en el módulo fuente es mayor que las especificadas en el módulo terminación, entonces al final de la simulación quedarán algunas entidades en el sistema.

### **Travesía de Entidades**

Las entidades efectúan su travesía entre los componentes del sistema por medio de ramas que los conectan.

El movimiento de una entidad a través de las ramas puede tomar tiempo, en cuyo caso la entidad será demorada en su arribo a su punto de destino.

Una lista de características deseables para el módulo de travesía de entidades es:

Una propiedad deseable del módulo de travesía de entidades es la habilidad para especificar el *tiempo de demora* como una constante, una variable aleatoria, una variable del usuario, o una expresión que combine varios de estos elementos. También, dado que varias entidades pueden transitar la misma rama a velocidades diferentes (lo que puede ser representado por uno de los atributos de la entidad), el módulo de travesía debería permitir el hacer al tiempo de demora una función de un atributo de la entidad.

Otra capacidad deseable del módulo es la habilidad de *señalar el fin del tiempo de travesía* con la ocurrencia de un evento. Esto posibilitará esos tipos de entidad cuyo tiempo de terminación puede no ser posible de programar por adelantado.

En muchas aplicaciones prácticas, las ramas a través de las cuales transitan las entidades tienen "prácticamente" *capacidad* ilimitada; o sea, un número ilimitado de entidades puede transitar la misma rama simultáneamente. Ejemplos de esto son los clientes que van desde la puerta del banco al final de la cola de espera y los aviones volando entre dos ciudades. En ciertas otras oportunidades la capacidad de la rama puede ser limitada. Ejemplo de esto son los autos moviéndose en una calle estrecha y el equipo de manipulación de materiales circulando por el piso de una fábrica. Una capacidad dentro del mismo módulo de travesía en sí mismo u otros modos de controlar la capacidad de las ramas debería permitir estas previsiones.

Dado que, a veces, las entidades necesitan *transportistas* para transportarlas desde sus fuentes a sus destinos (tal como personas ingresando a autos o a ómnibus, camiones transportando materiales de construcción y partes siendo movidas por el equipo de movimiento de materiales), es deseable que se provea, a quien construye el modelo, con la habilidad de asignar transportistas a las



entidades y a sus rutas. Más adelante se darán más informaciones acerca de los transportistas de entidades.

Nótese que los escenarios posibles de manipulación de materiales pueden ser muy complicados y dependientes de la tecnología.

Esto es especialmente verdad en el dominio de la manufactura.

En este dominio se pueden utilizar contenedores en los cuales los materiales se mueven todos a la misma velocidad sincronizada.

Se pueden utilizar grúas de techo que tienen aceleración, desaceleración, movimiento de la grúa principal, movimiento del contacto y movimientos verticales del artilugio de aprensión de materiales.

Pueden ser usados vehículos guiados automáticamente (automatically guided vehicle (AGV)), los que pueden ser asignados o enviados a rutas al azar que pueden cruzarse.

En el caso de grúas múltiples y AGV, la capacidad de la rama limita el número y velocidad de las entidades que la transitan.

También en estas situaciones las velocidades de las entidades que van adelante limitan las velocidades de las entidades que van atrás.

Dadas estas complicaciones, las herramientas especializadas de simulación dedicadas al dominio de la manufactura son, generalmente, más fáciles de utilizar que las herramientas de simulación de propósito general, algunas de las cuales no proveen medios para modelar estos sistemas complejos de manipulación de materiales.

### **Selección de Ruta (rama)**

En muchas situaciones especiales, las entidades tienen rutas alternativas que pueden tomar.

Son ejemplo los autos llegando a una intersección, los clientes que llegan y eligen líneas de espera diferentes y los clientes al principio de la cola que eligen ventanillas diferentes si más de una está disponible.

Las propiedades deseables del módulo que representa el proceso de selección son:

El módulo debería permitir la *selección probabilística* de las ramas. O sea, debería ser posible asignar a cada rama alternativa una probabilidad de selección por parte de la entidad que arriba. Un ejemplo de aplicación de este tipo de selección es el caso de las partes que están dejando una estación de inspección de una fábrica. Un cierto porcentaje de las partes deberán continuar a la operación siguiente, otro puede ser mandado a rehacer y el resto de las partes pueden desecharse. Aquí los valores porcentuales indican las probabilidades asociadas con cada rama alternativa que una parte inspeccionada puede seguir.

El módulo debería permitir la *selección condicional* de las ramas alternativas, esto es, debería ser posible asignar una condición (una regla) a cada rama alternativa. Entonces una rama es tomada por la entidad que arriba si se cumple la condición asociada. Las condiciones pueden ser basadas en los valores de variables globales, los valores de un atributo de la entidad, o una expresión compleja. Un ejemplo del caso de variable global es el envío de autos que arriban en una dirección antes de cierta hora del día y cambiándola, en sentido contrario, después de esa hora. Un ejemplo del caso de valor de atributo es el envío de clientes a servidores diferentes dependiendo del tipo de servicio requerido (siendo representado el tipo de servicio requerido por el atributo de la entidad) Un ejemplo del caso de la expresión es la selección de una rama alternativa si la longitud de cada cola posible es mayor que un cierto límite máximo.

Dado que la *selección de diferentes colas* (líneas de espera) es un proceso de ocurrencia común, es deseable tomar precauciones para algún desvío condicional sobre la base del estatus de las colas alternativas que una entidad que arriba puede elegir (la más corta, la capacidad remanente mayor, selección cíclica o al azar, etc.) Otro proceso común es la selección de cuál cola servir si el servidor tiene la opción de servir más de una cola. Ciertos criterios como seleccionar la cola más larga, selección de una entidad de cada cola (por ejemplo en las operaciones de ensamble), selección cíclica, o selección al azar son típicos. La **Figura 4.1** muestra un escenario donde se aplican estos dos tipos de selección.



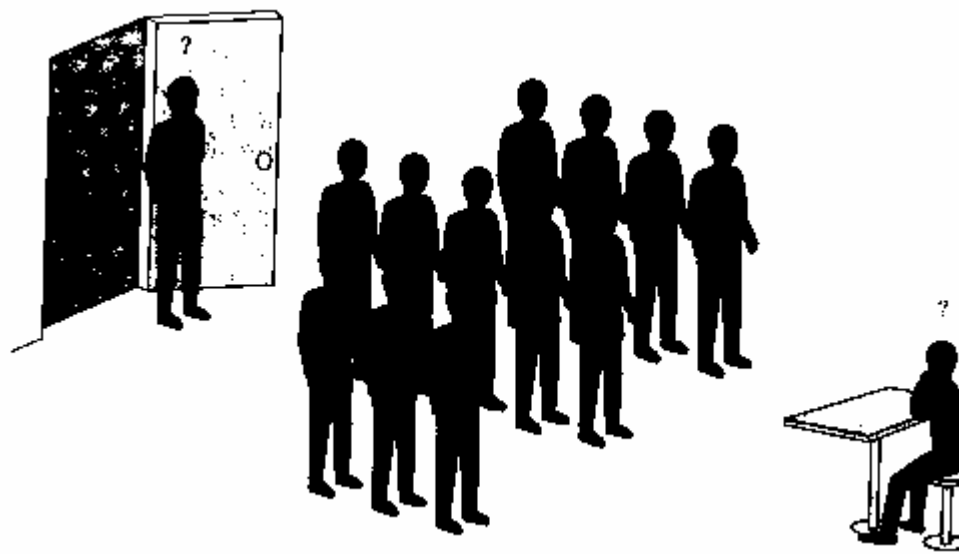


Figura 4.1

Selección de una cola donde ir y selección de una cola para servir

Otro proceso común es la *selección de un servidor* de entre múltiples servidores con características variables (ver **Figura 4.2**). Aquí los criterios posibles pueden ser la selección del servidor con el mayor período de ocio, selección cíclica, o selección al azar.

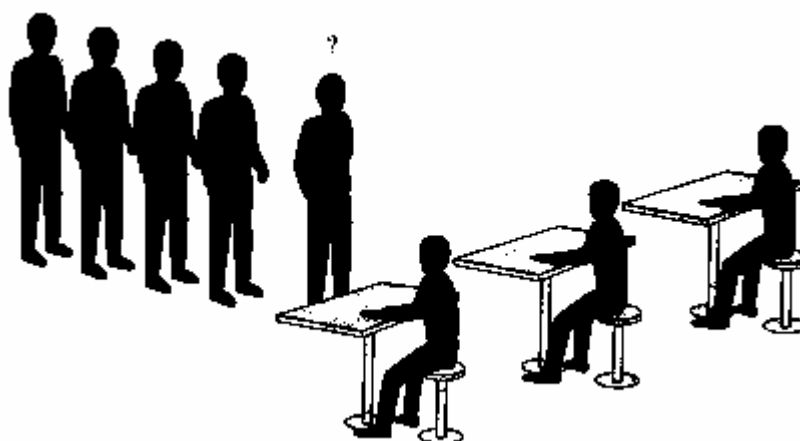


Figura 4.2

Selección de entre varios servidores

### Multiplicación y Reducción de Entidades

En una gran cantidad de sistemas del mundo real, las entidades simples se desagregan en un número mayor y varias entidades se juntan para formar una única entidad. Hay varias posibilidades en los procesos de multiplicación y reducción. A continuación se muestra una clasificación de esas posibilidades y las propiedades asociadas deseadas con un módulo de modelado de simulación que las representa:

Las entidades simples pueden ser multiplicadas para formar varias. Los ejemplos son las entidades agrupadas (batched entities) que están siendo desagrupadas (unbatched) en parte o en su totalidad para liberar las entidades originales que conforman el agrupamiento. Ejemplos específicos de esta clase de multiplicación son los pasajeros saliendo de un ómnibus, un ensamblaje siendo desensamblado y un documento simple siendo reproducido con cada copia tomando varias rutas en una red de transacciones de la oficina.

Las entidades pueden ser reducidas en número al *agruparlas (grouped)* para formar una entidad simple que puede ser desagregada en un estadio posterior en las entidades originales que conforman el grupo. Aquí las entidades originales pierden sus identidades (atributos) y la entidad agrupada puede tomar en sí los atributos de una de las entidades originales o una combinación de los atributos de todas las entidades originales. Por ejemplo, 5 cargas de pala de tierra pueden constituir una carga de camión de tierra en la cual las paladas individuales no son separables (y no necesitan serlo).

Las entidades pueden ser agrupadas para formar un agrupamiento que puede ser desagrupado (*unbatched*) en un estadio posterior. El grupo se trasladará como una entidad simple con atributos asignables. Las entidades originales, sin embargo, preservan sus identidades y pueden ser desagrupadas más tarde, tomando cada una una ruta diferente en el sistema. Ejemplos son las entidades pasajero ingresando a un ómnibus y partes siendo colocadas en un contenedor para su transporte o almacenado. Como los ejemplos de más arriba demuestran, los transportistas en algunos casos se pueden necesitar para mover las entidades agrupadas. Podría haber varias condiciones de carga para los transportistas (podrían ser llenados hasta su capacidad máxima, transportar lo que esté disponible, esperar cierto tiempo para partir, etc.), los transportistas pueden tener varias prioridades de partida, y pueden tener características diferentes (capacidades, rutas, velocidades, etc.).

Generalmente existen ciertas *condiciones de agrupamiento* (*grouping conditions*) que se aplican a cada uno de los escenarios de agrupamiento de más arriba. Estas condiciones generalmente están basadas o en el número de las entidades originales o en alguna combinación de valores de ciertos atributos de esas entidades. Por ejemplo, un bus de turismo puede partir cuando haya 40 pasajeros disponibles, o un camión puede partir cuando el peso total de carga disponible alcance su capacidad remanente. En el primer caso, el número total de entidades es la condición para el agrupamiento, en el último caso es el peso total individual, el cual se puede almacenar en cierto atributo de las entidades originales que fija la condición de agrupamiento.

Finalmente, las entidades pueden ser ensambladas para formar una única. Al revés de los casos de agrupamiento señalados más arriba, donde varias entidades provenientes de varias ramas pueden satisfacer la condición de agrupamiento, las entidades específicas, cada una proveniente de una cierta ruta generalmente se necesitan para completar una operación de ensamble. Ejemplo de esto son los componentes que se ensamblan para formar un subensamble y los documentos necesarios para formar un conjunto a ser enviado por correo.

### Uso de Recursos por Parte de la Entidad

Los recursos son bienes limitados que las entidades pueden utilizar por ciertos lapsos.

Generalmente, las entidades pueden moverse físicamente a ciertas ubicaciones donde están los recursos para utilizarlos.

Pueden también recibir otras clases de recursos que se les hacen disponibles donde estén las entidades.

El último tipo de recurso puede ser llevado por la entidad, utilizado, o consumido total o parcialmente, en cuyo caso una porción del recurso se puede liberar al inventario en algún estadio posterior.

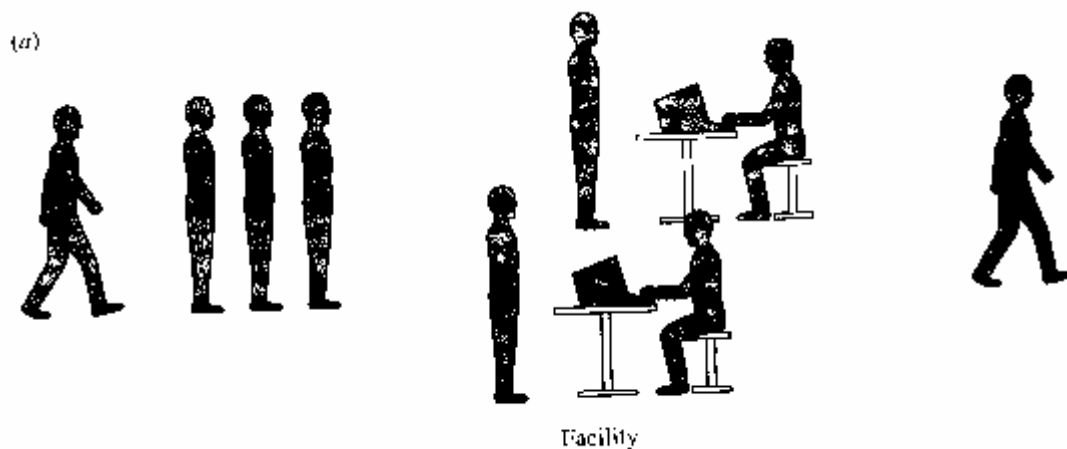
Dadas las diferencias entre los dos tipos de recursos, muchos lenguajes de simulación prefieren manejarlos en forma diferente.

Nos referiremos al primer tipo de recurso con el nombre especial de *instalación* (*facility*) y llamaremos al segundo tipo *recurso* (*resource*).

Las propiedades comunes de cada tipo de recurso son:

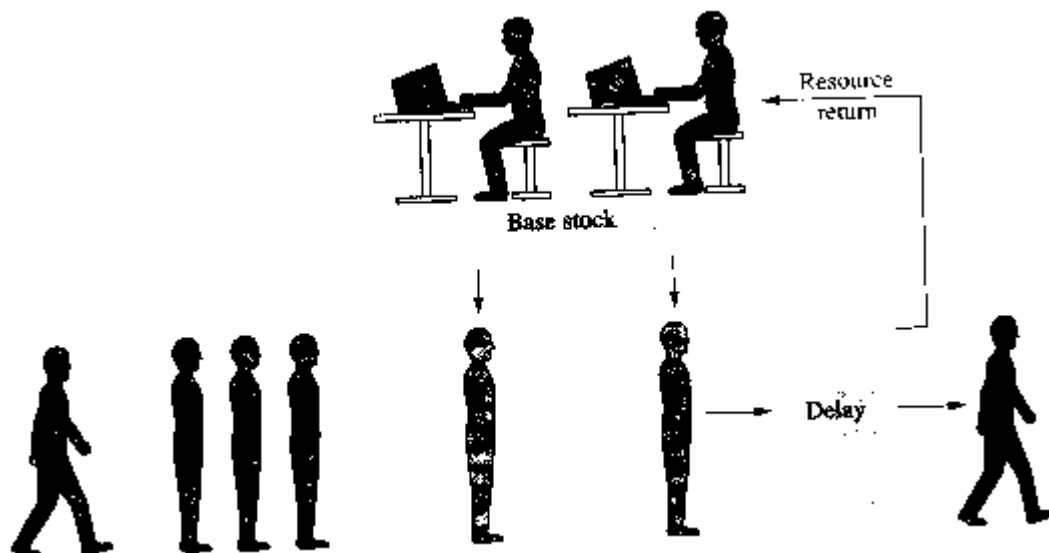
- Las instalaciones, usualmente, representan servidores, operadores, puestos de trabajo, o máquinas con los cuales son asociados ciertos *tiempos de servicio* (*service times*). La duración del servicio podría tener propiedades similares a aquellas del tiempo de travesía mencionado antes (fijo, variable aleatoria, etc.). Las facilidades pueden tener varios canales homogéneos trabajando en paralelo. Una propiedad común en una instalación es la posibilidad de detención o un suplemento por tiempo ocioso, que puede ser al azar o programado. Como se ha mencionado antes, ciertos criterios de selección pueden utilizarse cuando se selecciona de entre un grupo de facilidades no homogéneas o cuando se selecciona una entidad para atenderla de entre múltiples colas delante de una instalación que ha quedado libre. Finalmente, las entidades pueden necesitar concurrentemente otros tipos de recursos mientras reciben servicio en la instalación (tales como herramientas u operarios para una operación de maquinado).
- Los recursos generalmente tienen un *inventario* (*base stock*) que contiene inicialmente un número conocido de unidades de recurso. Las entidades pueden solicitar un cierto número de unidades de recurso. Puede existir una demora involucrada en la recepción del recurso. Los recursos tomados por la entidad pueden ser conservados mientras la entidad va a través de varios procesos (por ejemplo procesos de demora). Pueden ser devueltos al inventario después de usarse, en cuyo caso puede existir un tiempo de retorno. Las entidades pueden tener diferentes prioridades en el acceso a los recursos. En ciertos momentos una entidad de más alta prioridad puede quitar recursos a una de menor prioridad que está haciendo uso del recurso necesario. El tiempo durante el cual la entidad utiliza el recurso no necesita generalmente ser identificado dado que las entidades que poseen los recursos pueden ir a través de varias actividades que insumen tiempo en el sistema (tales como ser demoradas en actividades de travesía e insumir tiempo en las facilidades).

La **Figura 4.3** muestra dos escenarios para el modelado de una instalación con dos servidores paralelos.



(b)

Figura 4.3



La forma alternativa de representar instalaciones como recursos

En el primer escenario se utiliza una instalación en la cual están definidos dos servidores en paralelo.

Las entidades que arriban reciben servicio en la instalación por uno de los servidores.

La entidad es demorada en la instalación por la duración del servicio.

En el segundo escenario de modelado se define un inventario.

El inventario es inicializado para contener dos unidades del recurso.

Las entidades que arriban esperan la unidad de recurso una vez que ingresan al sistema.

Después de tomar el recurso son demoradas por la duración del servicio.

Después de este período de demora, el recurso es liberado a su inventario.

Debería hacerse un marcado énfasis en este punto con referencia a la opción de quien idea el modelo de identificar un componente del sistema como o una entidad o un recurso.

Hablando en general, cuando el componente de un sistema es representado como una entidad, el modelador tiene la opción de identificar rutas específicas para el movimiento de la entidad.

Esta posibilidad no existe para recursos que se extraen de su inventario, se usan y se regresan al inventario.

Las rutas para el movimiento del recurso no son identificables salvo que una entidad se especifique para llevar el recurso a través de alguna ruta específica.

Por ejemplo, considere la posibilidad de representar una pieza de equipamiento de manipulación de materiales como un recurso.

En tal caso un transportista ocioso puede suponerse que esté en algún lugar no identificado (inventario) para el cual el tiempo de viaje hasta la ubicación de la entidad puede ser especificada con una cierta demora, pero la ruta exacta para la travesía del transportista puede no estar identificada.

Si, por otro lado, el transportista es identificado como una entidad, uno puede especificar la ruta exacta para llegar a la entidad que solicita el transportista.

El equipamiento de manipulación de materiales es tratado como un recurso en algunos lenguajes de simulación.

Dependiendo de los escenarios específicos y de la información deseada del estudio de simulación, esto puede ser considerado una limitación!

Otra distinción entre entidades y recursos es que cada entidad individual puede ser trazada a través del sistema, pero los recursos del mismo tipo no pueden ser caracterizados individualmente (las unidades de recurso no pueden asumir atributos!). Esto lleva al hecho de que las unidades individuales de recurso en el inventario son indistinguibles. O sea, uno puede no saber, por ejemplo, qué unidad específica de recurso arribó en último término al inventario.

A diferencia del inventario de recurso que mantiene un número simple representando la cantidad de recursos disponibles, los puntos de acumulación de entidades (colas) son más complejos en su estructura dado que debe mantener registro de la información acerca de cada entidad individual (atributos de la entidad y orden de la entidad en la cola) contenidos en ellos.

Una ventaja de usar recursos en lugar de entidades (cuando es posible) es que los recursos demandan menos memoria de computadora.

Consideremos un ejemplo que clarifica el punto.

Supóngase que en una operación de manufactura los componentes A y B se ensamblan juntos con tornillos.

Una alternativa es representar los componentes y los tornillos como entidades, como muestra la **Figura 4.4a**.

(a)

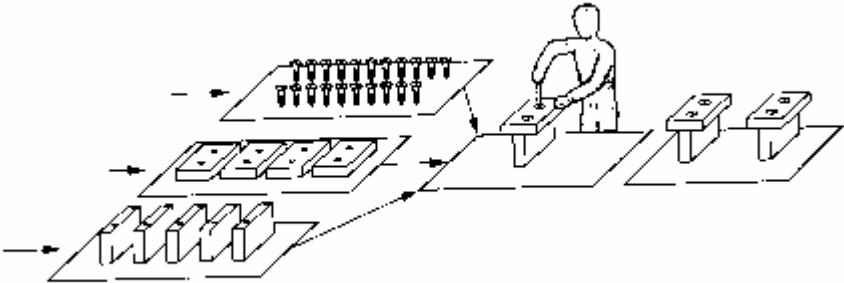
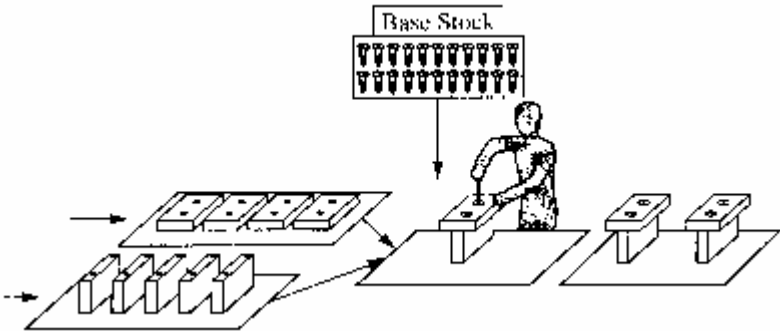


Figura 4.4

El modo alternativo por el cual algunas entidades son representadas como recursos (ver junto con **Figura 4.4b**)

Sin embargo, sabiendo que los tornillos son materiales de uso en el taller (shop-floor usage — (SFU)) y que montones de ellos están disponibles en los lugares de ensamble, no son enviados individualmente a los lugares vía una ruta particular, es más eficiente definirlos como recursos con un inventario. Estas unidades de recurso pueden ser tomadas en el lugar de ensamblado para completar el ensamble. La **Figura 4.4b** muestra este escenario alternativo.



Nótese que al revés del ejemplo anterior, aquí los recursos no retornan al inventario. Note también que en este ejemplo algunas unidades de recurso (no necesariamente una) puede ser tomada para cada operación.

**Acumulación de entidades**

Como se ha dicho, a través de sus movimientos a lo largo del sistema, las entidades pueden esperar en ciertos puntos de acumulación por muchas razones. En los sistemas discretos estos puntos de acumulación se llaman generalmente *colas* (queue). Primero identificaremos las circunstancias donde son necesarias las formaciones de colas y luego haremos una lista de las propiedades que, potencialmente, se aplican a todas las colas, sin importar el tipo de situación que hace necesaria su existencia.

Los procesos en los que habitualmente se forman colas son:

- La situación más común de formación de colas ocurre *cuando las entidades requieren la utilización de instalaciones (when entities require the use of facilities)*. Las instalaciones pueden estar ocupadas o temporariamente no disponibles por razones diferentes al hecho de estar procesando entidades (rotura, mantenimiento programado, detención del operador, etc.). Las entidades que

arriban y hallan la instalación que necesitan no disponible o ocupada tienen que esperar en una cola. Puede haber colas múltiples (paralelas) frente a una instalación. Por ejemplo, en una operación de ensamblado, cada uno de los varios componentes que dejan su operación de maquinado o el almacén tienen que esperar en sus colas respectivas frente al puesto de ensamblado. La instalación podría necesitar entonces esperar a recibir un componente para realizar el ensamble. Note que la disponibilidad de la instalación, en este caso, es una condición necesaria, pero no suficiente, condición para la partida de la entidad de la cola. La disponibilidad de las otras entidades que emparejan es la otra condición. Como se ha explicado, varias otras reglas de selección se aplican a las instalaciones que tienen colas múltiples.

- La formación de colas puede ocurrir cuando las entidades que arriban *requieren un cierto recurso* (*request a certain resource*) (diferente a la instalación). Si el inventario de recurso se vacía, las entidades deben hacer cola hasta que el recurso esté disponible. Cada entidad puede requerir una o más unidades de recurso. Por lo tanto, una entidad que tiene sólo una porción de su recurso disponible debe esperar en la cola hasta que se provea el número requerido de recursos.
- *El agrupamiento de entidades* (*entity grouping*) es otra condición que necesita la formación de colas. Cuando cierto número de entidades se necesita para formar un grupo (tanto en forma de batch o combinada), todas las entidades que arriban antes de la última (que completa el número requerido) deben aguardar en una cola antes de partir bajo la forma de una entidad agrupada única. También, cuando se utilizan transportistas para mover un grupo de entidades, las entidades que van arribando deben formar una cola para aguardar el arribo del transportista.
- En ciertas situaciones las entidades deben *esperar el arribo de otras entidades* (*await the arrival of other entities*) con el mismo valor de un atributo específico en ciertos puntos antes de que puedan continuar. Una vez que las entidades que casan (*matching*) están disponibles en sus colas asignadas, todas las entidades que casan salen en forma simultánea, cada una tomando su propia ruta. Nótese que la diferencia principal entre el proceso de ensamblado y el de casado es que en el proceso de ensamble las entidades que arriban se combinan para formar una sola entidad mientras que en el proceso de casado se demoran en colas sólo para ser sincronizadas con las entidades con que van a casar. En la instancia de casado cada entidad puede tomar una ruta separada y moverse a una velocidad diferente. Como un ejemplo de formación de colas para casado de entidades, considere el caso de descargar un camión que distribuye cajas de gaseosas a los almacenes. Cierta número de obreros (entidades) toma las cajas del camión, las lleva a la puerta del almacén y se las entrega a otros obreros que las toman y las apilan en el almacén. En esta situación, si cualquiera de los empleados que tanto entregan como reciben no está disponible, el otro debe aguardar a la entrada del almacén. Una vez efectuada la entrega los obreros toman sus propias rutas.
- Finalmente, las entidades en ciertos puntos del sistema pueden tener que *pedir permiso* (*await permission*) para proceder. El permiso puede ser otorgado por otras entidades en ubicaciones remotas del sistema. Ejemplos son los autos esperando en una intersección la luz verde, los clientes esperando la apertura de la entrada de una exposición, y los documentos esperando una firma antes de seguir con otras transacciones de la oficina. El punto de partida para las colas formadas por las razones referidas se llama puertas (*gates*) en algunos lenguajes de simulación. Asociados con las puertas hay interruptores (*switches*) que pueden abrir y cerrar las puertas. Los interruptores pueden ser activados o por las entidades que atraviesan las puertas o por otras entidades transitando en lugares distantes de la puerta. Aunque es posible utilizar timers para abrir y cerrar las puertas con la frecuencia deseada, una característica deseable del módulo representando el proceso de la puerta es un auto—timer para la puerta que logra su apertura y cierre automáticos con ciertas duraciones. Esta característica facilita el trabajo de modelado y elimina la necesidad de la creación de entidades imaginarias para activar los interruptores de la puerta. Ejemplos de aplicación de esta característica son el timing de los semáforos y la apertura y cierre de la puerta de entrada en un cine al comienzo de cada proyección (?).

Con las colas se pueden asociar las siguientes propiedades generales:

- Las colas pueden tener una *capacidad del área de espera* (*waiting area capacity*) limitada o ilimitada para acomodar las entidades que arriban. Ejemplos de áreas limitadas son el número de asientos disponible en el área de espera de un peluquero, el espacio disponible para autos en un cajero automático para utilizar desde el auto, el buffer entre dos operaciones consecutivas de maquinado, el espacio en un almacén, el buffer de memoria de una computadora para programas (*jobs*) que ingresan, y cosas por el estilo. Cuando una entidad que arriba encuentra una cola llena, puede rebotar y regresar a la cola después de cierto tiempo, o puede ir a otra parte del sistema. Alternativamente, la entidad que arriba puede bloquear el movimiento de las entidades detrás de ellas, pero esto sólo puede ocurrir cuando las entidades precedentes pueden ser acomodadas por un componente del sistema. Un proceso común de bloqueo es un grupo de máquinas dispuestas en serie con buffers limitados entre ellas. Cuando un buffer se llena la entidad en la instalación precedente puede ser forzada a permanecer en la instalación hasta que haya espacio disponible, aún si su tarea en la instalación ha sido completada. Esta situación produce el bloqueo de la instalación.



- Las colas pueden tener varias disciplinas de prioridad (*priority disciplines*) para ordenar entidades. Primera en llegar, primera en salir (*first-in-first-out*, o *FIFO*), que se aplica a situaciones tales como clientes en una línea de espera de un banco, y último en entrar, primero en salir (*last-in-first-out*, o *LIFO*), que se aplica a casos como las partes apiladas antes de una operación de maquinado, son dos disciplinas comunes de prioridad en colas. Son también comunes otras disciplinas que pueden estar basadas en el valor del atributo de una entidad. Ejemplos de estas disciplinas son las situaciones en las cuales el orden de los clientes o trabajos se fija de acuerdo a que primero se atiende a los que necesitan menos tiempo de procesado. La edad de los clientes o el tiempo de procesado de un trabajo puede ser representado por un atributo. La disciplina de prioridad en estos ejemplos será en base al valor más alto o más bajo de un atributo dado, respectivamente.
- Las entidades que arriban a una cola pueden tener un limitado tiempo de espera aceptable después del cual pueden optar por abandonar la cola y dirigirse a otro punto del sistema. Ejemplos son los clientes que utilizan un cierto servicio de una compañía telefónica que pueden colgar después de haber esperado un cierto tiempo, clientes esperando en una cola que deciden irse dado que la cola avanza poco, y un sistema de interruptores de telecomunicaciones que selecciona un nuevo canal después de cierta demora para dirigir una transacción que está esperando un canal libre
- Las colas pueden tener condiciones iniciales específicas. Pueden estar inicialmente vacías o no vacías. Este es un fenómeno común en los sistemas cíclicos que contienen un número limitado de entidades que dejan la cola, van a otras partes del sistema, y regresan a la cola para comenzar todo el proceso de nuevo. Un ejemplo de este proceso es el caso de camiones que esperan ser cargados en un área minera, viajan hasta el punto de descarga y regresan a la cola original para ser recargados. Nótese que las colas que están inicialmente no vacías pueden actuar como fuentes de entidades. Es deseable tener la opción de definir y fijar los valores de los atributos para las entidades disponibles inicialmente en la cola.

### Operaciones Auxiliares

Hasta ahora, nuestra discusión de procesos comunes en sistemas discretos ha considerado esas ocurrencias que corresponden a fenómenos físicos fácilmente observables en el mundo físico.

Para cada uno de los procesos descritos, uno puede construir o utilizar un módulo de modelado (llamado, generalmente, un objeto, una primitiva o un constructo) que representa el proceso.

Cuando uno construye un modelo de un sistema hay, sin embargo, ciertas operaciones auxiliares que se pueden necesitar, las cuales pueden o no corresponder a algún fenómeno físico.

A continuación se describen algunas de las más comunes de las operaciones auxiliares.

### Operaciones de Asignación

En muchas situaciones quien realiza el modelo puede necesitar definir algunas variables globales o atributos de las entidades y asignar, u ocasionalmente cambiar, los valores de esas variables o atributos. Por ejemplo, para rastrear el número total de entidades en una sección del sistema, uno puede definir una variable global e incrementarla cada vez que una variable entra a la sección y decrementarla cada vez que una entidad deja la sección.

De manera similar, un atributo de entidad representando una característica, digamos peso, puede cambiar en algún lugar del sistema (tal como el peso de un camión que ha volcado su carga).

También se puede hacer un cambio en el nivel corriente de un recurso, o en cierto momento se puede fijar en un nuevo nivel.

Los valores que están afectados pueden ser cambiados en montos constantes o al azar o pueden ser derivados de complejas expresiones.

A veces es deseable condiciones si—entonces—de lo contrario (*if-then-else*) en un módulo simple de asignación para evitar excesivas bifurcaciones condicionales.

Esta capacidad será útil en situaciones donde una entidad debe ser dirigida a uno de entre varios puntos alternativos de asignación en base a un conjunto dado de condiciones.

En general, cuanto más capacidades se construyan dentro del módulo de asignación de la herramienta de simulación orientada a procesos, más flexible se vuelve la herramienta para modelar en una variedad de situaciones.

### Operaciones de Archivo

En ciertos casos surge la necesidad de transferir, eliminar, o copiar entidades que ya están en cola.

Nos referimos a estas transacciones como operaciones de archivo (*file operations*).

La elección del término archivo (*file*) es debido al hecho de que las colas son estructuras de datos complejas en las cuales las entidades con varios atributos se guardan en un cierto orden para su futuro procesado.

El indicador para selección de entidades en archivos generalmente es el rango de la entidad en el archivo (primero, último, o enésimo).

Un atributo de entidad específico que tenga un valor específico es otro indicador común para la selección de entidades en archivos.



Los clientes que dejan una cola larga para unirse a una más corta o un trabajador postal retirando correspondencia de tercera clase del correo entrante para ser procesada son ejemplos de operaciones de archivo.

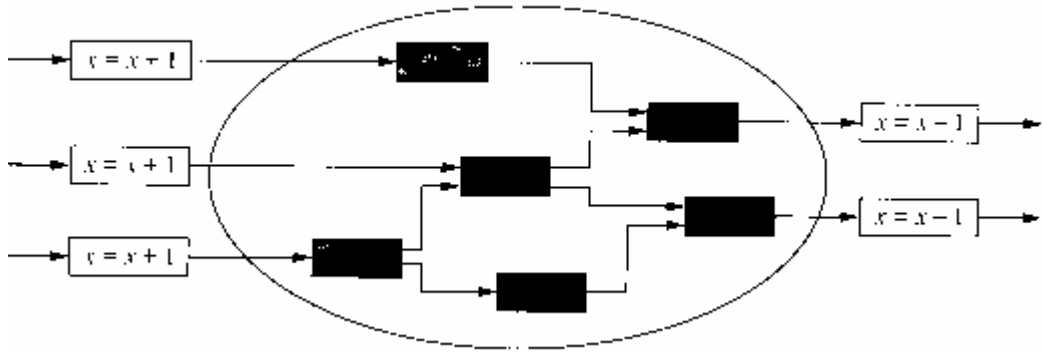
**Manipulación y Utilización de Variables**

Típicamente, hay dos clases de variables globales que se usan con frecuencia en los estudios de simulación de sistemas.

Estas son las variables del usuario (*user variables*) y las variables del sistema (*system variables*).

Las variables del usuario, como su nombre indica, son definidas explícitamente por el usuario para propósitos específicos.

Por ejemplo, como se muestra en la **Figura 4.5**, una variable de usuario puede ser definida para rastrear el número total de entidades en una cierta sección del sistema.



**Figura 4.5**

La variable *x* del usuario es utilizada para mantener el rastro del número total de entidades en cierta sección del sistema

En este caso, cada vez que una entidad entra a la sección incrementa la variable, y cada vez que una entidad deja el sistema decrementa la variable (suponiendo que las entidades no se multiplican ni dividen dentro del límite considerado).

Las colecciones de estadísticas basadas en el tiempo pueden proveer la información deseada.

Las variables del sistema son aquellas variables que son creadas automáticamente por la herramienta de simulación, y su número y tipo depende del modelo específico utilizado.

Las variables de sistema más comunes son:

- tiempo real de la simulación;
- número real de las entidades en varias filas de cola;
- número real de las entidades en las instalaciones;
- número real de los recursos utilizados y así por el estilo.

Las variables del sistema se usan frecuentemente cuando se especifican condiciones, duraciones, y varias formas de expresiones, y en varios estadios de asignación para fijar variables del usuario y atributos de entidades.

Una herramienta de simulación debería proveer acceso a estas variables cuando lo solicite el usuario.

**Inicialización del Sistema**

Las variables del usuario, los niveles de recursos, el estatus de las puertas y el número posible de entidades existentes en colas y los valores correspondientes de los atributos deberían ser identificados al comienzo de la simulación.

Una herramienta de simulación debería permitir la inicialización de estas cantidades antes de la ejecución real del modelo.

**Especificación de las Estadísticas**

Otra operación auxiliar en un modelo de simulación es la especificación de las estadísticas deseadas.

Un modelo de simulación, sin importar cuan sofisticado y preciso sea, puede ser inútil si no provee la información deseada acerca de la performance de varios aspectos del sistema.

Ciertas estadísticas que en general necesitan los usuarios se refieren a la longitud de las colas, los tiempos de espera de las entidades en las colas, la utilización de las instalaciones y la utilización de recursos.

Dada la gran frecuencia con que los usuarios necesitan estas estadísticas, es deseable que una herramienta de simulación las genere automáticamente.

Hay, sin embargo, otras clases de estadísticas (específicas del caso) que deberían dejarse que las especifique el usuario.

Son generalmente de uno de los siguientes tipos:

- Estadísticas acerca de los tiempos de arribo de entidades a un punto del sistema.
- Estadísticas acerca del número de entidades que pasan a través de un cierto punto del sistema (conteo de entidades).
- Estadísticas acerca del tiempo de travesía de las entidades entre dos puntos del sistema, siendo la más común el tiempo del sistema, o el tiempo total que ha pasado en el sistema (desde la creación hasta la terminación).
- Estadísticas sobre el tiempo entre el pasaje de entidades sucesivas desde un cierto punto del sistema.
- Estadísticas sobre variables globales.

Las cantidades generales deseadas en los informes estadísticos son la media, el desvío estándar, el mínimo, el máximo y el último valor observado al final de la simulación.

También es deseable ver gráficos de frecuencia, histogramas y /o gráficos de las estadísticas sobre un período determinado y con las especificaciones deseadas de los gráficos tales como tamaño de las celdas o tiempos entre impresión de gráficos.

Como vimos, las estadísticas en sistemas dinámicos son de dos tipos, basadas en la observación y basadas en el tiempo.

Una característica deseable de una herramienta de simulación es la previsión de que se pueda especificar varias estadísticas y sus tipos.

La herramienta de simulación debería trazar apropiadamente, computar y sacar en un formato legible las estadísticas especificadas.

Dado que en muchas instancias los analistas necesitan estudiar la performance de sus sistemas bajo condiciones de estado estable, es deseable descartar las estadísticas durante el período transitorio (período de arranque en frío) de los sistemas bajo estudio de simulación.

La herramienta de simulación debería permitir al usuario el definir el largo estimado después del cual deben ser recolectadas las estadísticas.

También es una característica deseable (porque puede ahorrar tiempo de experimentación del modelo) de una herramienta de simulación la capacidad de realizar automáticamente corridas múltiples, cada una de ellas con un conjunto posible y diferente de condiciones iniciales (valores diferentes de los parámetros, niveles iniciales de los recursos, valores iniciales de las variables del usuario, etc.).

Esta característica puede no ser crucial si la herramienta provee facilidades para una rápida modificación y ejecución del modelo (por ejemplo, si operaciones como modificación del código fuente y compilación no son necesarias para cada cambio en las condiciones iniciales).

### **Trazado de Entidades en el Modelo**

Los modelos de simulación son, generalmente, difíciles de revisar para corregir errores, verificar el modelo y verificar algunos aspectos básicos de la validación del modelo.

Consecuentemente, para ayudar a este proceso de verificación, los lenguajes de simulación proporcionan generalmente la capacidad de trazar las entidades para mostrar la posición de ciertas entidades especificadas en varios estadios de la ejecución del modelo.

Un modelo que corre sin errores no es necesariamente un modelo correcto, dado que pueden existir ciertas representaciones incorrectas de la realidad.

Una capacidad de trazado que imprima el nombre de la posición de las entidades especificadas en ciertos momentos del evento, o, con preferencia, una capacidad de animación que muestre el movimiento de las entidades en la red del modelo o el propio diagrama de flujo, es una característica esencial de una buena herramienta de simulación.

### **Animación de la Escena**

La animación de la escena es una capacidad que puede realzar la presentación de un modelo de simulación.

Una animación de escena generalmente muestra un fondo con ciertos componentes fijos (por ejemplo, máquinas en una simulación de fábrica, o cielos y montes en una simulación de vuelo) y algunos iconos, representando las entidades, moviéndose.

También pueden utilizarse ciertas primitivas gráficas en animación como gráficos de torta y de barra, medidores y diales para mostrar ciertas estadísticas de interés.

Algunas herramientas de simulación ofrecen animación concurrente, otras animación de playback.

Una animación de tipo concurrente puede correr simultáneamente con el modelo de simulación, pero el tipo playback puede correr solamente después de que el modelo de simulación ha corrido y su resultado de la traza de ciertos hechos dentro del lapso deseado ha sido guardado en un disco.

En este caso el módulo de animación lee los datos del disco y cambia de una imagen de animación a otra de acuerdo con los cambios en los datos.

La animación generalmente muestra un período corto de la simulación del sistema, por ese motivo no es un medio apropiado para analizar la performance del sistema simulado.

En la longitud típica de una corrida de simulación ocurren muchos hechos que no pueden ser enteramente capturados en unas pocas imágenes de animación.

Además, la creación de escenas de animación requiere un esfuerzo considerable, por lo que la animación de escenas se usa raramente para depuración y trazado.

A causa de la dificultad involucrada en construir escenas de animación y de la limitación de sus aplicaciones, la mayoría de los estudios de simulación no utilizan animación de escena.

Una aplicación importante de la animación de escena del tipo concurrente es en el entrenamiento de operadores (entrenamiento de pilotos utilizando un simulador de vuelo).

Otra aplicación importante de este tipo de animación es en los juegos de computadora.

Las propiedades deseables de la animación de escenas incluyen instalación de uso (preferiblemente sin programación), soporte de gráficos de alta resolución, una biblioteca de iconos de animación, operación a alta velocidad, movimiento suave de las imágenes móviles y soporte tridimensional.

La última propiedad raramente es incorporada en los paquetes de simulación comercial.

La mayoría de los paquetes de animación son bidimensionales, aunque algunas de sus aplicaciones parezcan tridimensionales.

Los sistemas verdaderamente tridimensionales están basados generalmente en estructuras de alambre (una representación bastante rústica).

La animación de escenas se utiliza principalmente para propósitos de presentación, pero una vez que se ha desarrollado, el trabajo de animación puede ser utilizado para los propósitos de verificación y validación del modelo.

Sin embargo, utilizar este tipo de animación sólo para el último propósito no es aconsejable, dado que, en general, insume mucho tiempo.

## **Modelado con EZSIM**

### **Objetos y Capacidades**

En este acápite se mostraran los conceptos generales descriptos utilizando el entorno EZSIM.

Aquí enfatizaremos el modelado de la simulación.

Los archivos de datos correspondientes a los ejemplos están incluidos en el disquete de EZSIM.

#### **Nodo Fuente**

El nodo SOURCE crea entidades.

Cada entidad que cree el nodo SOURCE puede tener un nombre, el que se torna un atributo para la entidad (este atributo es llamado NAME).

Si se desean más atributos para la entidad, se puede utilizar un nodo ASSIGN siguiendo al nodo SOURCE.

El nombre del atributo puede utilizarse para bifurcar, para fijar tiempos específicos de demora y para identificar la entidad con un símbolo dedicado en la animación.

Si no se especifica un nombre para la entidad se mostrará con un ? en la animación.

Es posible controlar el número de creaciones en el nodo SOURCE.

También las creaciones pueden ser comenzadas o finalizadas después de cierto tiempo.

Nótese que el detener la creación de entidades no finaliza la simulación.

La simulación puede continuar hasta que todas las entidades dejen el sistema, un nodo TERMINATE detenga la simulación, o el tiempo de simulación alcanza una longitud prescrita (especificada en el menú de Control).

Como se dijo, varios elementos de tiempo como creación de la primera entidad, tiempo entre creaciones, y tiempos de demora en los nodos DELAY y FACILITY tienen una ventana estándar que permite la especificación del tiempo como una constante, una distribución estadística, una variable del usuario o como una expresión conteniendo muchos elementos.

Cuando se especifica la elección de distribución estadística, una ventana con un listado de varios nombres de distribuciones permite la elección de la distribución deseada.

Después de que es especificada la distribución, el usuario es incitado con mensajes que le piden identificar los valores de los parámetros de la distribución y la elección de la semilla del número pseudoaleatorio a ser utilizado para el proceso específico.

#### **Nodo Terminar**

El nodo TERMINATE termina el recorrido de las entidades.

Cuando una entidad entra a un nodo TERMINATE se la considera fuera de la red, y la memoria de la computadora asignada a sus atributos se libera.

Cada nodo TERMINATE puede finalizar la simulación entera si el conteo de entidades especificado es alcanzado.

#### **Nodo Demora**

El nodo DELAY se utiliza para crear una demora que corresponde al tiempo de travesía de la entidad de un nodo al otro.

El tiempo de travesía se selecciona desde la ventana estándar de tiempo y puede ser representado por una constante, varias distribuciones estadísticas, un valor de atributo, una variable del usuario, o una expresión.

Los nodos DELAY pueden ser también usados para coleccionar varias ramas incidentes y para crear varias ramas que salen.

En tales casos el tiempo de demora puede ser fijado en cero.

**Ejemplo 5.1**

Este ejemplo demuestra una situación de múltiples ramas incidentes.

Supongamos que en la simulación de la autopista del **Ejemplo 3.1** hay una entrada adicional conectada en el medio de la autopista.

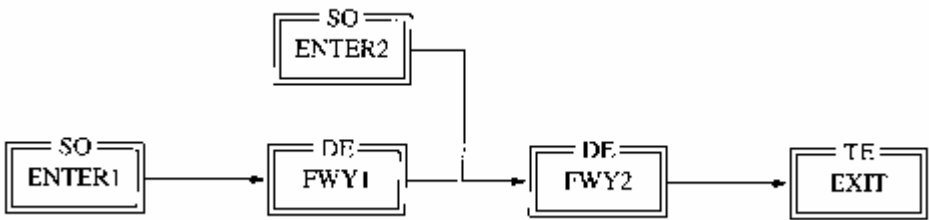
Supongamos además que el tiempo entre arribos de los autos que entran a la autopista está distribuido exponencialmente con media de 5 min

Los tiempos de viaje a través de los segmentos primero y segundo de la autopista (entre la primera y la segunda entrada y entre la segunda entrada y el final) están ambos distribuidos normalmente con media de 7, 5 min y desvío estándar de 1,5 min

Estamos interesados en simular el sistema para observar el número total de autos en cada segmento de la autopista.

Note que se está considerando sólo una dirección de la autopista (un lado de la autopista).

La red EZSIM para el modelo de simulación se muestra en la **Figura 5.1**.



**Figura 5.1**

**Demostración de múltiples ramas incidentes**

Los nodos SOURCE llamados ENTER1 y ENTER2 representan cada entrada de la autopista.

Usted puede especificar nombres diferentes (digamos, CAR1 y CAR2) y elegir símbolos de animación diferentes (digamos, 1 y 2) para los autos que arriban a través de la primera y segunda entrada, respectivamente.

Los nodos DELAY llamados FWY1 y FWY2 representan los tiempos de travesía a través de los dos segmentos de la autopista.

Siga el procedimiento descrito en los ejemplos **Ejemplo 3.1** y **Ejemplo 3.2** para construir la red e ingresar los parámetros nodales.

Corra el modelo en modo animación y observe el flujo de entidades en su modelo.

También observe los cambios en el número de entidades en cada nodo DELAY mientras el modelo es simulado.

**Nodo de Instalación**

El nodo FACILITY actúa como un servidor.

Las entidades permanecen en el nodo durante la duración de su servicio.

Cuando el nodo se ocupa, las entidades que arriban deben esperar hasta que el nodo esté libre.

**Un nodo FACILITY debe estar precedido por un nodo QUEUE.**

El nodo permite varios servidores paralelos.

Los servidores paralelos se supone que son homogéneos, o sea, tienen las mismas características.

En este nodo están disponibles provisiones como programación de interrupciones y uso simultáneo de varios recursos.

Varios criterios de selección para colas múltiples en paralelo pueden ser elegidos en el nodo FACILITY.

Por ejemplo, la opción ensamble (assembly — ASM) que junta varias entidades que ingresan es un criterio de selección.

La **Tabla 5.1** lista los criterios aplicables para las selecciones de instalación y cola y sus descripciones.

Esta tabla se puede ver mientras se está en el proceso de construcción presionando ALT+H cuando se utilizan elecciones de selección en el nodo FACILITY.

Se le puede dar un nuevo nombre a la entidad que deja el nodo facility.

Esto permitirá mostrar, por ejemplo, las partes procesadas y no procesadas con símbolos diferentes en el modo animación.

**Tabla 5.1**

Criterios de selección para instalaciones y colas

Condición	Descripción
<b>Aplicable tanto a instalaciones como colas</b>	
POR	Priority given in the order of node appearance in network. Prioridad dada en el orden de aparición del nodo en la red.
CYC	Priority is given in a cyclic manner - transfer to first available node after each selection. La prioridad se da de manera cíclica — transfiere al primer nodo disponible después de cada selección.
RAN	Priority given randomly - assign equal probabilities to nodes. Prioridad dada al azar — asigna probabilidades iguales a los nodos
LAV	Priority given to the node that has had the largest average number of entities in it to date. Prioridad dada al nodo que ha tenido el mayor número promedio de entidades en él a la fecha.
SAV	Priority given to the node that has had the smallest average number of entities in it to date. Prioridad dada al nodo que ha tenido el menor número promedio de entidades en él a la fecha.
<b>Aplicable a instalaciones</b>	
LRC	Priority given to the node with largest remaining unused capacity. Prioridad dada al nodo con la mayor capacidad no utilizada.
SRC	Priority given to the node with smallest remaining unused capacity. Prioridad dada al nodo con la menor capacidad no utilizada.
LBT	Largest amount of usage (busy time) to date. Mayor utilización (tiempo ocupado) a la fecha.
SBT	Smallest amount of usage (busy time) to date. Menor utilización (tiempo ocupado) a la fecha.
LIT	Select the node with longest idle time. Seleccione el nodo con el mayor tiempo ocioso.
SIT	Select the node with smallest idle time. Seleccione el nodo con el menor tiempo ocioso.
<b>Aplicable a colas</b>	
LWF	Priority given to the node that has had the longest waiting time. Prioridad dada al nodo que ha tenido el mayor tiempo de espera.
SWF	Priority given to the node that has had the shortest waiting time. Prioridad dada al nodo que ha tenido el menor tiempo de espera.
LNQ	Priority given to the queue with largest number of entities in it. Prioridad dada a la cola con el mayor número de entidades en ella.
SNQ	Priority given to the queue with smallest number of entities in it. Prioridad dada a la cola con el menor número de entidades en ella.
ASM	Select one entity from each queue and assemble them into a single entity. Seleccione una entidad de cada cola y ensámblelas en una sola entidad.

**Nodo Cola**

El nodo QUEUE representa buffers antes de los nodos FACILITY y siempre es seguido por un nodo FACILITY.

Una QUEUE puede tener una limitación de capacidad, en cuyo caso las entidades que arriben deben rebotar a un nodo específico, bloquear la entidad previa (si la hay) o terminar en el nodo.

El último caso puede ocurrir cuando la opción de bloqueo está seleccionada y no hay nodo FACILITY detrás del nodo QUEUE.

Cada vez que se especifica una capacidad limitada de cola, EZSIM automáticamente incita al usuario para que especifique las opciones de rebote o bloqueo.

Cuando se ha elegido la opción de rebote, EZSIM lista los nombres de todos los nodos de la red a los cuales se les permite entrada directa de entidades.

El tiempo de travesía de rebote puede ser también especificado usando la ventana estándar de tiempo.

Es posible especificar un tiempo de espera máximo de la entidad, después del cual la entidad puede rebotar a un nodo específico.

Las colas pueden tener varias disciplinas.

Pueden ser FIFO, LIFO o prioridad dada a las entidades con más alto o más bajo valor de un atributo dado.

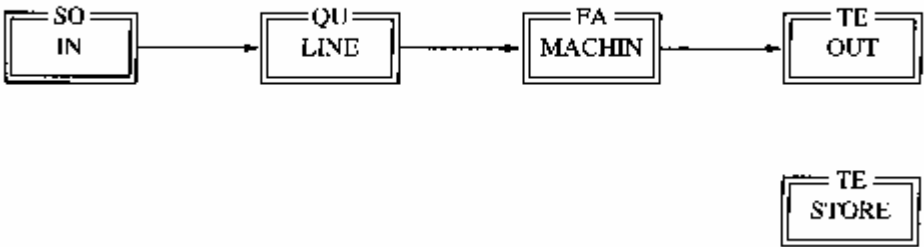
Las entidades abandonando la cola (o porque está llena o porque se ha alcanzado el límite de tiempo de espera) se mueven a su nodo de destino especificado a través de un camino invisible en la animación.



Hay disponibles muchos criterios de selección para las entidades que eligen a qué cola paralela unirse y para las facilidades eligiendo de cuál cola paralela recibir entidades.  
Cuando un nodo QUEUE está conectado a varias facilidades que lo siguen, varias selecciones de una instalación están disponibles, como se ha listado en **Tabla 5.1**.

**Ejemplo 5.2**

En este ejemplo, se demuestran algunas de las capacidades de los nodos QUEUE y FACILITY.  
Suponga que grandes partes llegan a un puesto de maquinado para su procesado.  
El tiempo entre arribos de las partes es 2 min  
El buffer anterior a la instalación puede acomodar un máximo de 5 partes.  
Las partes que arriban y encuentran el buffer lleno se envían a un área de almacenaje temporal.  
El tiempo de maquinado está distribuido exponencialmente con media de 1.5 min  
La máquina se detiene cada 30 min para un ajuste de 5 min durante el cual las partes no pueden ser procesadas.  
El procesado de cada parte requiere 10 tornillos.  
20.000 tornillos están inicialmente disponibles en existencia.  
Estamos interesados en simular el sistema durante la generación de 100 partes para hallar las estadísticas de la cola y de la instalación, el tiempo total que las partes procesadas están en el sistema, y el número total de partes no procesadas que se envían al área de almacenaje.  
La **Figura 5.2a** muestra el diagrama de red del sistema.



**Figura 5.2<sup>a</sup>**

Demostración de los nodos QUEUE y FACILITY

En el estadio de especificación de parámetros del nodo QUEUE se indica una capacidad limitada de 5.  
Se elige la opción rebotar (*balking*) para las entidades que llegan y encuentran la cola llena.  
El nodo TERMINATE llamado STORE representa el destino de esas entidades.  
El tiempo de travesía de rebote puede ser fijado en cero.  
Todos los otros campos del nodo QUEUE se fijan en sus valores por defecto.  
Dado que el camino de las entidades rebotadas a su destino es transparente, el nodo STORE no está conectado a ningún otro nodo.  
En el nodo FACILITY, se especifica una parada programada con la frecuencia y duración especificada.  
También se especifica una necesidad de utilización de recurso.  
Al recurso se le da el nombre SCREW y al incitador de pregunta pidiendo el número de recursos necesarios se contesta ingresando el número 10.  
No se especifican retornos de recursos.  
Todas las otras preguntas en la ventana de recurso se satisfacen con selecciones de valores por defecto.  
Después que el usuario especifica los parámetros de todos los nodos y presiona ESC, EZSIM conoce que se utiliza un recurso SCREW en el modelo, pero que no se ha identificado su nivel inicial.  
Por lo tanto, pregunta al usuario el nivel de recurso por medio de una ventana especial de inicialización.  
Después del estadio de inicialización, aparece la ventana de estadísticas.  
Las estadísticas en la cola y en la instalación son recolectadas e informadas automáticamente por EZSIM.  
Para obtener las estadísticas del tiempo total del sistema de las partes procesadas, se define SYSTIME en el menú de estadísticas, para el cual el tiempo de travesía entre los nodos IN y OUT son especificados de una manera similar a la del **Ejemplo 5.1**.  
Para obtener el número total de partes no procesadas que han sido enviadas al área de almacenamiento se ingresa un nombre, digamos UNPROCD en la primer columna de la ventana de estadística.  
Luego se elige el tipo de estadística (la cual en este caso es conteo de entidades).  
Finalmente, en la lista de nodos que EZSIM provee, se selecciona STORE, significando que deseamos obtener el conteo de entidades en ese nodo.  
La **Figura 5.2b** muestra el output de simulación para 1.000 unidades de tiempo.  
Para crear esta salida, se elige la opción Print Output en el Output menú.



```

*** E Z S I M  STATISTICAL  REPORT  ***

Simulation Project: EX2
Analyst: BK
Date: 4/2/92
Disk file name: EX2.OUT

Current Time: 1000.000   Transient Period: 0.000

Q U E U E S:
NAME      MIN/MAX/LAST      MEAN      STD      MEAN      STD
          LENGTH           LENGTH    LENGTH    DELAY    DELAY
-----
LINE      0/ 5/ 4           1.53      1.58      3.04      3.75

F A C I L I T I E S:
NAME      NBR      MIN/MAX/LAST      MEAN      STD      MEAN      MEAN
          SRVRS    UTILIZATION    UTLZ      UTLZ      IDLE      BUSY
-----
MACHIN    1         0/ 1/ 1         0.79      0.41      0.44      1.66

V A R I A B L E S:
NAME      MEAN      STD      MIN      MAX      No.OBSRVD
-----
SYSTIME   4.85E+00  4.25E+00  1.89E-02  2.06E+01  475
UNPROCD   2.10E+01  0.00E+00  2.10E+01  2.10E+01  21

R E S O U R C E S:
NAME      INIT      MIN/MAX/LAST      MEAN      STD      MEAN      STD
          LEVEL    USAGE           USAGE    USAGE    LEVEL    LEVEL
-----
SCREW     20000     0/4760/4760     2352.28   1374.15   17647.72  1374.15

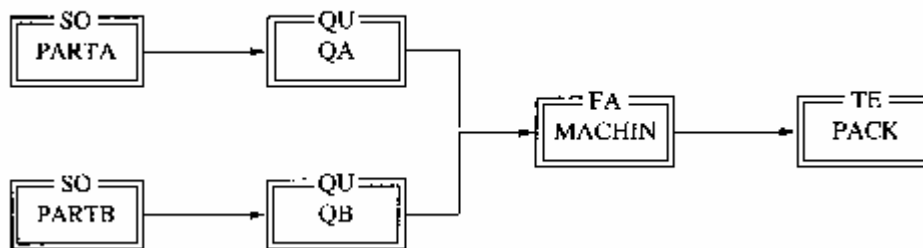
```

Figura 5.2b

Salida para el modelo del Ejemplo 5.2

Ejemplo 5.3

Este ejemplo muestra un caso de selección de múltiples colas ante un servidor.  
Suponga que las partes A y B arriban, cada una de una ruta diferente, a una estación de ensamblado.  
El tiempo entre arribos está distribuido uniformemente entre 7 y 12 minutos.  
El tiempo de ensamblado es constante y dura 4 min para cada operación de ensamblado.  
Desearíamos simular la operación para encontrar las estadísticas en el producido de la instalación (el número de partes ensambladas por unidad de tiempo).  
La **Figura 5.3** muestra el diagrama de red del modelo.



**Figura 5.3**

Selección desde múltiples colas

Cada nodo SOURCE crea un tipo de parte.

Cada parte ingresa a su propio buffer antes del servidor.

En el estadio de identificación de parámetros en el nodo FACILITY (una vez que se hayan respondido todas las preguntas), una ventana de selección de ramas salta pidiéndole al usuario que identifique el criterio para la selección desde las dos colas delante de la instalación.

Para ver la lista de los criterios disponibles, presione ALT+H.

La elección correcta en este caso es ASM, la cual toma una entidad de cada cola y las ensambla en una.

Para encontrar las estadísticas en el sistema completo, el tipo de estadística seleccionado en el menú de estadísticas debería ser "tiempo entre arribos al nodo".

Identificando PACK como el nodo para el cual se desean las estadísticas del tiempo entre arribos da un valor que es la inversa del paso a través de la instalación.

Para ver la lógica del ensamblado en operación, se recomienda la animación de este ejemplo.

#### **Ejemplo 5.4**

En este ejemplo se demuestra el caso de múltiples instalaciones con una cola simple.

Supongamos que a un lugar de construcción llegan camiones para ser cargados por uno de los tres cargadores.

Si hay más de un cargador libre en el momento del arribo del camión, un supervisor asigna el cargador con el tiempo ocioso más largo al camión.

Los tiempos entre arribos de camiones están distribuidos exponencialmente con una media de 1,2 min.

Cada cargador tiene un tiempo de servicio distribuido exponencialmente con medias de 2, 4 y 5 min respectivamente.

Estamos interesados en las estadísticas de la cola y las estadísticas del número total de camiones atendido por cada cargador.

También deseamos saber el número de camiones que dejan el sistema luego de 1.000 min de operación.

Note que dado que estas instalaciones en paralelo tienen características de servicio diferentes, deben ser identificadas como facilidades separadas en el modelo; de lo contrario, un nodo FACILITY con 3 servidores en paralelo habría representado el sistema.

La red para este ejemplo se muestra en la **Figura 5.4**.

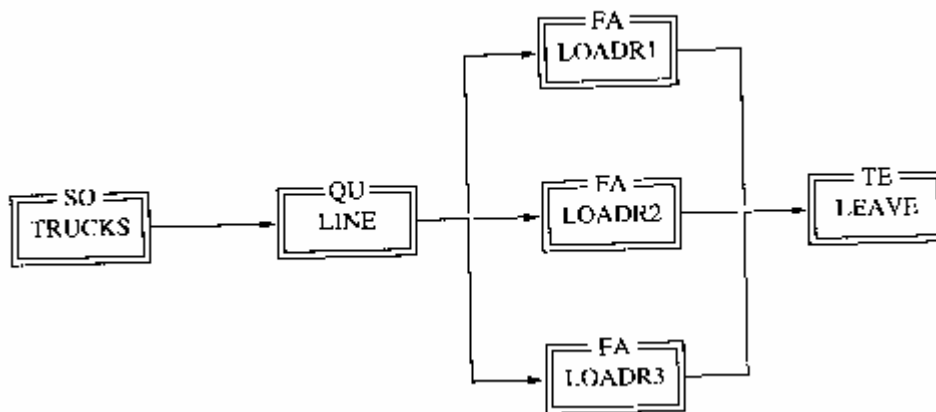


Figura 5.4

Ilustración de selección de entre facilidades

En el estadio de identificación para el nodo QUEUE, se mostrará una ventana de selección de rama.

Una opción dentro de esta ventana es Bifurcación selectiva.

La elección de esta opción resultará en la aparición de otra ventana que muestra la lista de opciones de selección de facilidades.

La opción LIT (mayor tiempo ocioso — *largest idle time*) es identificada como el criterio de selección de instalación.

Después de completar la red, se identifican las estadísticas deseadas, las que están en la forma de conteo de entidades en los 4 nodos (LOADR1, LOADR2, LOADR3 y LEAVE).

Mirando este modelo en animación demuestra la forma en que se seleccionan los cargadores.

### Ejemplo 5.5

En este ejemplo se demuestra el uso de bifurcación probabilística.

Supóngase una versión simplificada del **Ejemplo 5.2**, donde la cola tiene capacidad infinita.

Supongamos que el 80 % de las partes que dejan la instalación tendrán una calidad aceptable, y el resto debe ser enviado de nuevo a la cola de maquinado para hacer nuevamente el trabajo.

La **Figura 5.5** muestra la red del modelo de esta situación.

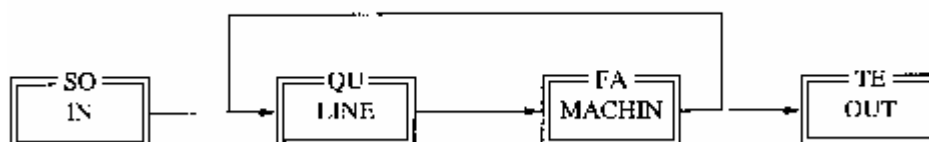


Figura 5.5

Ilustración de bifurcación probabilística

En esta red el nodo FACILITY está conectado al nodo QUEUE.

(Nótese que si la conexión se hace después de la identificación del nodo FACILITY está completa, entonces al momento de la conexión de este nodo cambiará de un borde de línea doble a uno de línea simple. Esto significa que se requieren aclaraciones adicionales.)

Cuando se selecciona el nodo FACILITY para identificación de parámetros, aparecerá una nueva ventana para especificar el criterio de bifurcación.

En esta ventana se proveen varios tipos de criterios.

Ellos son los criterios de bifurcación Probabilístico, Condicional, Última elección y Siempre.

La opción probabilística permite la asignación de probabilidades de asignación mutuamente excluyentes a cada una de las ramas emergentes.

El valor total de estas probabilidades debería ser uno.

(EZSIM rechaza inputs que violan esta regla regresando el cursor a la posición de punto decimal del número que viola la regla.)

En este ejemplo, se selecciona la opción Probabilística para las ramas emergentes del nodo FACILITY.

Se especifican los valores 0,8 y 0,2 para las ramas que llevan a los nodos TERMINATE y QUEUE, respectivamente.

Observe este modelo en animación mientras las partes rehechas dejan la instalación y se unen al fin de la cola.

### Ejemplo 5.6

En este ejemplo se demuestra la bifurcación condicional.

Esta opción permite especificar varias condiciones aplicables a cada una de las ramas que emergen de un nodo.

Las condiciones pueden estar basadas en los valores de

- a) variables del sistema (tales como tiempo de simulación, número de entidades en una cola dada, o valor de un recurso disponible);
- b) atributos de una entidad que está por viajar a través de la rama; y
- c) variables del usuario.

Pueden ser construidas expresiones complejas para identificar las condiciones en las ramas.

Supongamos que un taller recibe dos tipos de partes para procesarlas en un día determinado.

El tiempo entre arribos de la parte Tipo1 está distribuida exponencialmente con una media de 4 min, la parte Tipo2 arriba cada 5 min.

La política del taller es procesar siempre partes Tipo1.

El taller de maquinado completa las piezas en un tiempo que se distribuye exponencialmente con media 5 min.

Si la cola antes de la estación de maquinado tiene menos de 3 partes, entonces una parte Tipo2 será enviada a la cola para ser procesada.

Aquellas partes Tipo2 que arriban dentro de las dos primeras horas de operación y que no se han acomodado en el taller se envían a un subcontratista para ser procesadas y devueltas antes del fin del día.

Si una parte Tipo2 arriba después de las dos primeras horas de operación y no está procesada en el taller, se envía a un área de almacenaje.

Estamos interesados en el número de partes que son procesadas en el taller, subcontratadas y enviadas al almacén durante un día de operación.

La **Figura 5.6a** muestra el modelo de red del sistema.

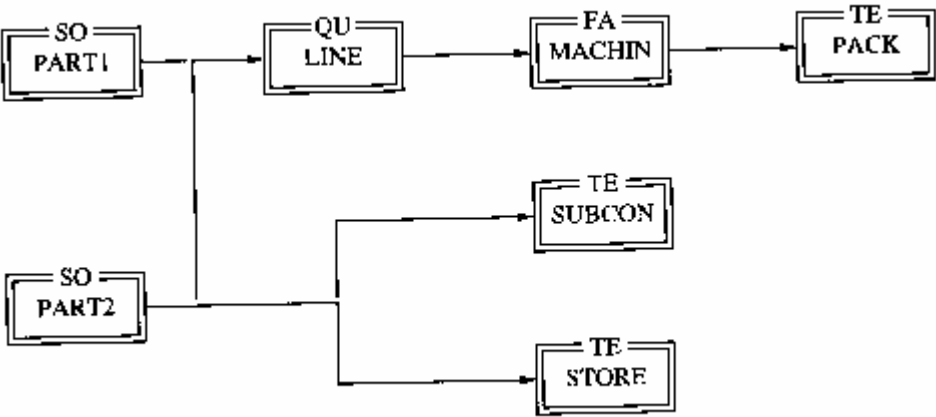


Figura 5.6a

Ilustración de bifurcación condicional

El nodo fuente PART1 genera las partes TIPO1 y las envía a una cola denominada LINE. Entonces se mueven hacia el lugar de maquinado. El nodo fuente PART2 genera las parte Tipo2, las cuales pueden ir a la máquina para su maquinado, al subcontratista, o a un área de almacenamiento. Las condiciones son especificadas para cada rama usando la opción Bifurcación Condicional. Para especificar la condición de la rama a la cola, se selecciona la opción Variable del Sistema. Entonces se selecciona la Longitud de la Cola de la ventana de opciones disponibles. Dado que hay una sola cola en el modelo, EZSIM provee sólo una elección de opción, que es LINE. Tan pronto como se identifica una cola específica, aparece LEN(LINE) en el lado izquierdo de la expresión de condición que indica la longitud de la cola llamada LINE. Se presenta una ventana mostrando los operadores relacionales. Se selecciona < y se ingresa el número 3. La **Figura 5.6b** muestra la ventana para la bifurcación condicional en el nodo PART2.

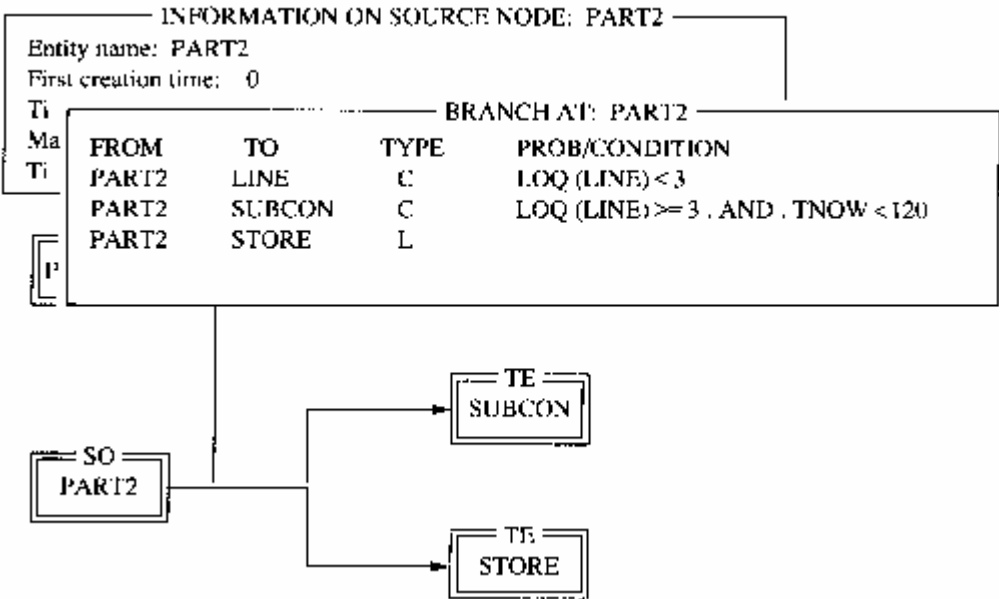


Figura 5.6b

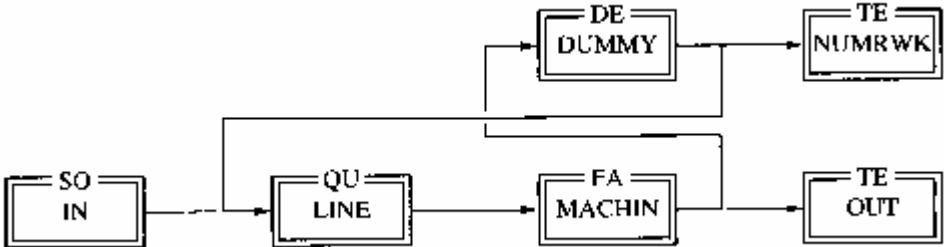
La ventana para condiciones de bifurcación

Nótese que la segunda expresión utiliza un operador AND, que combina las dos expresiones. TNOW (el tiempo de la simulación) puede ser seleccionado en esta expresión seleccionando primero la opción Variable del Sistema y luego eligiendo Tiempo de Simulación de la ventana correspondiente. La tercera condición utiliza la bifurcación Última Elección. La opción de bifurcación Última Elección especificada para una rama determinada resulta en la selección de la rama cuando ninguna otra rama puede ser tomada. El número máximo de ramas que pueden ser tomadas debería ser identificadas cuando estén especificadas todas las condiciones. En este caso, este número es 1. Si se seleccionan números mayores a 1, puede ocurrir una multiplicación de entidades (una entidad será enviada a través de cada rama cuya condición es satisfecha).

Por supuesto, esto ocurrirá sólo si más de una condición es satisfecha simultáneamente.  
Si ninguna de las condiciones es satisfecha, EZSIM destruye la entidad.  
En el modo de animación el usuario es advertido acerca de la eliminación de entidad con un mensaje en la parte inferior de la pantalla.  
En el menú de estadísticas, los conteos de entidades son especificados en los nodos PACK, SUBCON y STORE.  
Corriendo la simulación por el lapso deseado provee las estadísticas de outputs para estos conteos, como se ha especificado para el problema.  
Se recomienda primero correr este modelo en el modo animación para observar el pasaje de entidades en varias ramas mientras el contenido de la cola y el valor del tiempo de simulación cambia.

**Ejemplo 5.7**

Este ejemplo demuestra el uso de la opción Siempre de bifurcación.  
La especificación de esta opción sobre una rama resulta en que se toma la rama sin importar el estado de las otras ramas.  
Si simultáneamente se toman otras ramas, la opción de bifurcación Siempre puede resultar en duplicación de la entidad.  
Para demostrar esta opción, suponga que en el **Ejemplo 5.5**, estamos interesados en encontrar el número total de partes que se envían a reprocesar.  
La **Figura 5.7** muestra la red para este modelo.



**Figura 5.7**

Ilustración de la opción de bifurcación Siempre

Puede ser especificado un nodo ficticio DELAY con duración cero al que se envían las unidades a ser reprocesadas.  
El propósito de adicionar este nodo es convertir una rama emergente del nodo FACILITY en dos ramas.  
Las dos ramas emergentes conectan el nodo DELAY con los nodos LINE y NUMRWK, respectivamente.  
El último nodo recibe una entidad ficticia duplicada por cada transacción de reprocesado.  
La opción de bifurcación Siempre se especifica para cada una de las dos ramas que salen del nodo DELAY.  
Se especifica conteo de entidades en la elección de estadísticas para los nodos OUT y NUMRWK en el menú de estadísticas.

**Nodo Asignar**

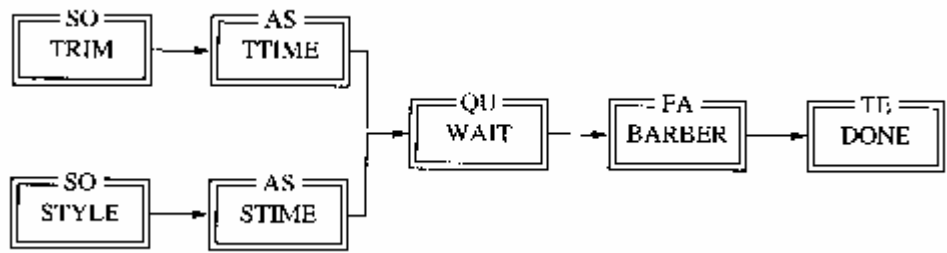
El nodo ASSIGN puede ser usado para fijar nuevos atributos o cambiar los valores de los atributos de las entidades que pasan a través del nodo.  
El nodo también puede ser usado para fijar los valores de variables definidas por el usuario y para fijar nuevos niveles o incrementar o decrementar los niveles actuales de recursos.  
También hay disponibles en este nodo provisiones para el uso de condiciones y expresiones.

**Ejemplo 5.8**

Este ejemplo muestra el uso del nodo ASSIGN para fijar atributos de las entidades.  
Suponga que a una peluquería llegan clientes que requieren dos tipos de servicios: recorte y peinado.  
Suponga que los clientes con cualquiera de los dos tipos de requerimiento de servicio, o sea recorte o peinado, llegan a la peluquería.  
Los tiempos entre arribos de clientes que necesitan recorte y peinado se distribuye exponencialmente con medias de 15 y 24 min respectivamente.  
Los tiempos de recorte y peinado también están distribuidos exponencialmente con medias de 10 y 20 min, respectivamente.  
Hay dos peluqueros.  
Deseamos observar las estadísticas de las colas para decidir acerca del número de sillas a ser ubicadas en el área de espera.



La **Figura 5.8** muestra la red para el modelo de este sistema.



**Figura 5.8**

Uso del nodo ASSIGN para fijar valores de atributos

El nodo SOURCE crea las entidades clientes con cada tipo de requerimiento de servicio. La aproximación que utiliza el modelado es guardar los correspondientes requerimientos de tiempo de servicio en un atributo de cada entidad inmediatamente después de que ingrese al sistema. Esta información es llevada con la entidad a la instalación en la cual es utilizada como la duración del servicio.

En el nodo ASSIGN de TTIME, se selecciona la opción Atributo de la ventana de elecciones para asignación. Al atributo se le da el nombre SRVTIME (un nombre arbitrario), y su valor es especificado como un ejemplo de una distribución exponencial con media de 10 min.

En el nodo ASSIGN de STIME, se selecciona la misma opción (Atributo) y al atributo se le da el mismo nombre (SRVTIME). El valor del atributo es especificado como un ejemplo de una distribución exponencial con media 20 min.

En el nodo FACILITY se especifican dos servidores en paralelo. De la elecciones dentro de la ventana de duración del servicio, se selecciona Atributo. Tan pronto como se ha hecho esta selección, aparece una ventana con una lista de todos los nombres de atributos definidos hasta ese momento. Note que esta lista incluye SRVTIME y NAME. El último es un atributo por defecto que es definido internamente por EZSIM y lleva el nombre de la entidad.

Para especificar la duración del servicio, se selecciona SRVTIME. SRVTIME lleva la duración del servicio de la entidad. Note que cada entidad lleva su propia duración del servicio a la instalación por medio de su atributo.



Ejemplo 5.9

Este ejemplo demuestra una aproximación alternativa al modelado del sistema descrito en el ejemplo previo.

En lugar de almacenar el tiempo de servicio requerido en un atributo de las entidades cliente, se puede construir en el nodo FACILITY una tabla que asocia el "nombre" de la entidad al tiempo de servicio relacionado.

En este caso, cuando una entidad arriba al nodo, se observa su nombre de atributo y su tiempo de servicio asociado se extrae de la tabla.

La **Figura 5.9a** muestra la red para este modelo.

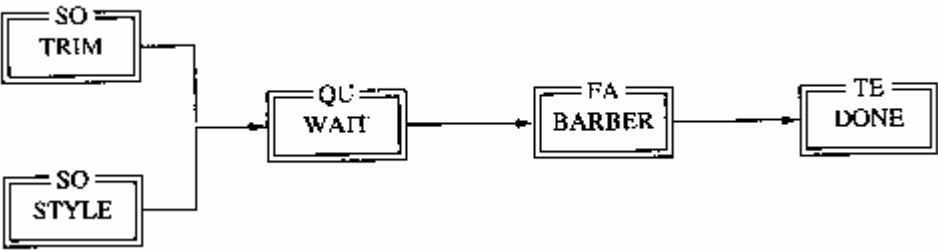


Figura 5.9a

Ilustración de tiempo de servicio dependiente del nombre

En cada nodo SOURCE, se les da un nombre a las entidades creadas.

En el nodo FACILITY se selecciona la opción tiempo de servicio dependiente de la entidad, y los tiempos de servicio relacionados son asignados a cada nombre de entidad.

La **Figura 5.9b** muestra la ventana en la cual se ha hecho la asignación.

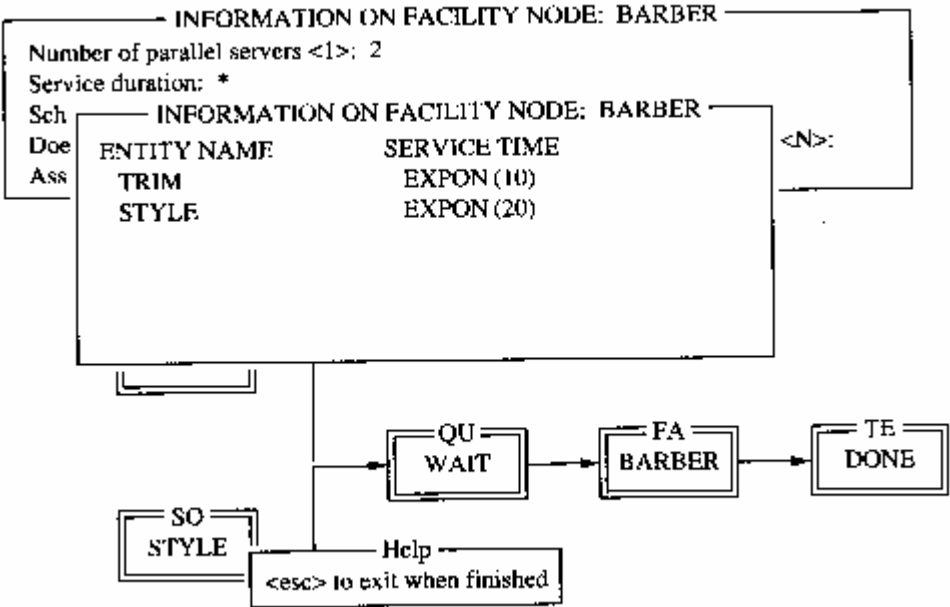


Figura 5.9b

Ventana para especificar tiempos de servicio

Nótese que relacionar duraciones de servicio con nombres de entidades en las instalaciones es útil en las aplicaciones de manufactura donde las máquinas procesan varias partes, cada una de las cuales tiene un requerimiento diferente de tiempo de procesamiento.

Ejemplo 5.10

Este ejemplo muestra el uso del nodo ASSIGN para manipular una variable del usuario.

Suponga que hay dos cajeros en un banco, cada uno con su línea de espera de clientes.

El primer cajero procesa depósitos y el segundo extracciones.

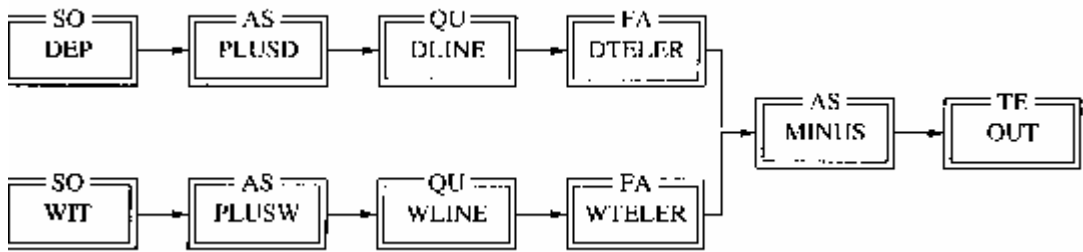
Los tiempos entre arribos de los clientes tienen una distribución exponencial con medias de 7 y 4 min para transacciones de depósito y extracción, respectivamente.

El tiempo de servicio para depósitos es uniforme con un rango de 2 a 4 min.

El tiempo de extracción es también uniforme, con un rango de 3 a 6 min.

Estamos interesados en las estadísticas sobre el número total de clientes en el banco.

La **Figura 5.10** muestra la red para este ejemplo.



**Figura 5.10**

Uso del nodo ASSIGN para fijar una variable del usuario

Los dos nodos ASSIGN, llamados PLUSD y PLUSW en este ejemplo, incrementan la variable del usuario, X, en una unidad cada vez que una entidad pasa a través de ellos.

El nodo ASSIGN llamado MINUS decrementa X en una unidad cada vez que una entidad pasa a través de él.

De esta manera, el valor de X siempre representa el número total de entidades que hay en el sistema.

Para fijar el valor de X en el nodo ASSIGN, se selecciona la opción Variable del Usuario del nodo.

A la variable se le da el nombre X.

Entonces se selecciona la opción de Expresión para fijar X ó en  $X + 1$  ó en  $X - 1$

En la salida del modo gráfico, EZSIM pide el valor inicial de la variable X Este valor se fija en cero.

En el menú de estadística, un nombre de variable — digamos NUMINSYS — puede ser dado a las estadísticas deseadas.

Entonces se selecciona la Variable del Usuario.

EZSIM provee entonces una lista de todas las variables definidas en el modelo (la lista contiene sólo X en este caso)

Se selecciona entonces la variable X.

Las estadísticas se pueden recolectar o como basadas en el tiempo (lo cual tiene más sentido) o basadas en la observación.

En el último caso, un nodo debería ser identificado como el punto de observación.

Cada vez que una entidad pega en un nodo, el valor de X se registra como una observación estadística.

Corra este modelo en el modo de animación, y mientras está en ese modo, presione la tecla V para ver el valor de X al ir cambiando a lo largo del tiempo.

### Nodo de Recurso—Q

El nodo RESOURCE—Q es un buffer en el cual las entidades esperan para recibir un número especificado de uno o más de recursos.

Cuando los recursos están disponibles en su almacenaje base, la entidad se libera y los niveles del inventario base de los recursos tomados se reduce en las cantidades correspondientes.

Las entidades que toman los recursos pueden liberarlos total o parcialmente en puntos específicos de la red.

En este nodo se pueden especificar también las prioridades para tomar recursos, los tiempos para recuperar recursos y regreso al inventario base.

Si varias entidades esperan en diferentes nodos RESOURCE—Q para tomar un recurso dado, entonces cuando un recurso se torna disponible, la entidad en el nodo con mayor prioridad toma el recurso primero.

La prioridad para tomar un recurso puede ser especificado en el nodo RECURSO—Q por un número entero (cero es para la prioridad más baja).

Obviamente, cuando un solo nodo RESOURCE—Q (o un nodo FACILITY que utiliza recursos) se utiliza en el modelo, el valor de prioridad es irrelevante y puede ser fijado en su valor por defecto (cero).

Algunas entidades pueden tener poder de requerir recursos.

Tales entidades pueden tomar unidades de recurso que están siendo utilizadas por otras entidades si las unidades en el inventario base no contiene el número necesario de unidades de recurso.

Las entidades cuyos recursos pueden ser tomados por entidades que pueden tomar recursos, pueden continuar utilizando los recursos cuando sean liberados por la entidad con poder de requerir recursos.

En EZSIM pueden ser especificadas las siguientes opciones de poder para tomar recursos:

1. Poder para tomar recursos que han sido tomados más temprano.
2. Poder para tomar recursos que han sido tomados en último término.
3. Poder para tomar todos los recursos en uso.
4. Poder para tomar todos los recursos (por ejemplo, en uso así como los del inventario de base).

Las opciones 1 y 2 se aplican en situaciones en las cuales varias unidades que están en uso han sido tomadas en diferentes puntos del tiempo, y la entidad con poder para tomar recursos tiene la opción de tomar su número requerido de recursos de entre los que han sido tomados más temprano o aquellos que han sido tomados más tarde.

La opción 3 permite a la entidad con poder de tomar recursos tomar todos los que están en uso, independientemente de su número.

Un ejemplo de aplicación es cuando un corte de energía corta todos los suministros de las máquinas en la fábrica.

La opción 4 permite a la entidad con poder de tomar recursos tomar todos los recursos en uso así como todas las unidades en el inventario base.

Un ejemplo de aplicación de esta opción es cuando una cinta transportadora cuyos segmentos son representados como recursos se descompone.

En este caso, no estarán disponibles todos los segmentos de la cinta transportadora, estén en uso o no.

**Ejemplo 5.11**

En un aeropuerto, arriban aviones de carga a una velocidad de uno por hora.

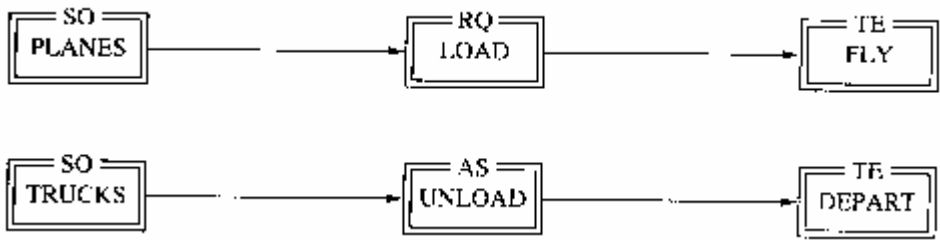
Cada avión espera mientras carga 70 unidades de carga y luego parte.

Los camiones, cada uno trayendo 10 unidades de carga, arriban a la terminal con un tiempo entre arribos distribuidos exponencialmente con una media de 9 min.

Inicialmente hay 100 unidades de carga en la terminal.

Deseamos simular el sistema para 50 aviones que arriban para encontrar el tiempo esperado de espera de cada avión antes de despegar.

La **Figura 5.11** muestra el diagrama de la red del modelo.



**Figura 5.11**

Ilustración de los nodos RESOURCE—Q y ASSIGN

Hay dos segmentos disjuntos en la red del modelo.

El segmento superior representa el movimiento de los aviones, y el inferior el de los camiones.

Cada avión que arriba va a un nodo RESOURCE—Q y pide 70 unidades de un recurso llamado CARGO (la especificación de parámetros para el nodo RECURSO—Q es similar al de la sección recursos del nodo FACILITY).

Tan pronto como un número suficiente de unidades de carga está disponible, el avión sale.

Las entidades camión van al nodo ASSIGN después de su creación.

En el nodo de asignación, ellos incrementan el nivel de recurso, CARGO, en 10 unidades, y continúan al punto de terminación.

Nótese que el inventario base del recurso es el elemento que liga los dos segmentos de red aparentemente no relacionados.

Corra el modelo en el modo animación, y presione V en la animación para observar el nivel del recurso mientras cambia en el tiempo.

**Ejemplo 5.12**

Este ejemplo muestra cómo un recurso puede especificarse para jugar un rol de instalaciones móviles.

Supóngase que en una manufactura automatizada textil, un operador está a cargo de controlar varias máquinas de hilado que pueden detenerse a causa de defectos del material.

Estos hacen que el operador deba involucrarse para resolver el problema.

El operador está ubicado en un cuarto de control donde un monitor indica la máquina específica que necesita atención y la naturaleza del problema a resolver.

Hay tres tipos de máquinas situadas en tres áreas diferentes.

El tiempo entre fallas del grupo de máquinas Tipo1 es Exp(40), y toma Exp(2) para resolver el problema.

Los tiempos para los grupos de máquinas Tipo2 y Tipo3 son Exp(30)/Exp(3) y Exp(45)/Exp(2), respectivamente.

Las unidades de tiempo son minutos.

Los tiempos de viaje del operador desde el cuarto de control a las áreas 1, 2 y 3 son 3, 2 y 4 minutos, respectivamente.

Después de cada reparación, el operador debería regresar al cuarto de control.

Deseamos simular este sistema para encontrar el nivel de utilización del operador y el tipo de detención promedio de cada tipo de máquina que resulta de la falta de disponibilidad del operario.

La **Figura 5.12** muestra el modelo de la red para el sistema.

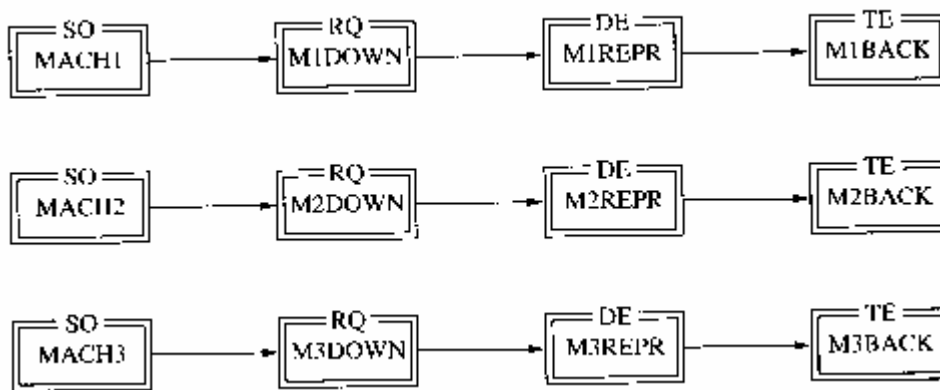


Figura 5.12

Ilustración de los recursos utilizados para representar instalaciones móviles

Note que cada nodo source crea detenciones de máquina para el grupo correspondiente de máquinas. Cada máquina que requiera la atención del operador va al nodo RESOURCE—Q para tomar un recurso llamado MAN.

El número requerido es 1, y los tiempos para obtener y retornar el recurso corresponden a los tiempos de viaje del operador desde el cuarto de control a las máquinas y regreso.

Después de cada nodo RESOURCE—Q hay un nodo DELAY que representa el tiempo de reparación asociado.

En el nodo RESOURCE—Q se especifica que el recurso tomado debe ser dejado después del nodo delay que sigue al nodo.

En el estadio de inicialización, el número inicial de recurso, MAN, es fijado en 1.

Note que las estadísticas de la cola para cada nodo RESOURCE—Q debe ser conservada e impresa independientemente por EZSIM, dadas las estadísticas deseadas por el enunciado del problema.

### Nodo Puerta—Q

El nodo GATE—Q contiene un buffer y una puerta detrás del buffer.

Cuando la puerta está abierta, las entidades en el buffer pueden dejar el nodo, de lo contrario deben esperar en el buffer hasta que la puerta se abra.

El estatus inicial de una puerta puede ser fijado tanto en abierto como en cerrado.

Las puertas pueden ser abiertas o cerradas por nodos SWITCH que pueden ser activados por algunas entidades remotas.

Cuando se abre una puerta, las entidades en su buffer pueden salir todas juntas.

Para controlar el número de entidades que salen en el momento de apertura de la puerta, se puede activarse un SWITCH que cierre la puerta después que la n—ésima entidad deja el nodo.

El nodo GATE—Q también tiene un timer construido en él que puede abrir y cerrar la puerta con el timing deseado.

El nodo GATE tiene su propio buffer, y por lo tanto no debería ser precedido por un nodo QUEUE.

Cada puerta puede ser inicializada para tener un estatus abierto o cerrado al inicio de la simulación.

### Nodo Interruptor

El nodo SWITCH puede ser utilizado para abrir o cerrar las puertas desde cualquier punto remoto de la red.

Cualquier entidad que pase a través del nodo en un momento dado puede activar el interruptor.

La ventana de identificación de parámetros del nodo SWITCH provee una lista de todas las puertas utilizadas en el modelo, y permite la especificación de las puertas que son abiertas o cerradas cada vez que una entidad pasa a través de un nodo SWITCH.

### Ejemplo 5.13

Este ejemplo muestra cómo un nodo GATE, con su auto—timer activado, puede ser utilizado para controlar el flujo de entidades.

Suponga que una demostración comienza cada 30 minutos.

Después del inicio de cada demostración, que toma alrededor de 15 minutos, la puerta del showroom se cierra por 20 minutos.

Después se abre por 10 minutos.

Las personas arriban para ver el show con un tiempo entre arribos uniforme que va de 0,8 a 1,2 minutos.

Para diseñar un área de espera a la sombra, estamos interesados en las estadísticas de la cola donde las personas están esperando por el show.

La Figura 5.13a muestra el modelo de la red para este sistema.





Figura 5.13ª

Ilustración de nodo GATE con timer

El nodo SOURCE genera el arribo de personas al nodo GATE, el cual representa el área de espera. La puerta se dispone que esté inicialmente abierta y tiene un timer que en forma alternativa la mantiene abierta por 10 minutos, la cierra, la mantiene cerrada por 20 minutos y la abre de nuevo. Las entidades que dejan este nodo se envían al nodo TERMINATE. Para observar las estadísticas de la cola, se pueden seleccionar las estadísticas de la cola en el menú de output. La **Figura 5.13b** muestra el equivalente del modelo de más arriba utilizando interruptores externos en lugar de un timer para abrir y cerrar la puerta.

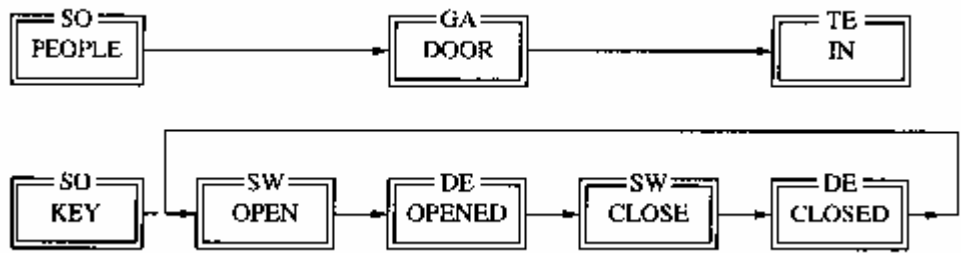


Figura 5.13b

Ilustración de un nodo GATE controlado por nodos SWITCH

El modelo está constituido por dos redes disjuntas. La red superior es para el flujo de las entidades personas. La puerta aquí no tiene un timer. La red disjunta inferior es para el flujo de una simple entidad imaginaria que abre y cierra la puerta con los timings deseados. Nótese que el nodo SOURCE debería ser especificado para una entidad solamente. Entonces esta entidad simple circula en el loop y abre y cierra la puerta mientras va a través de los nodos SWITCH llamados OPEN y CLOSE. Los nodos DELAY especifican los lapsos durante los cuales la puerta permanece abierta y cerrada. Mire este modelo en animación. Note que cuando abre la puerta, todas las entidades dejan la puerta al mismo tiempo de simulación. El nodo SWITCH se crea con W.

**Ejemplo 5.14**

Este ejemplo demuestra cómo un número controlado de entidades puede ser liberado desde una puerta. Suponga en el ejemplo previo que el showroom tiene una capacidad máxima de 45 personas. Al comienzo de cada show, la puerta se abre, y un máximo de 45 personas entra al showroom. Estamos interesados en las estadísticas del número de personas que no pueden entrar al showroom al comienzo del show. La red de la **Figura 5.14** muestra el modelo para este ejemplo.

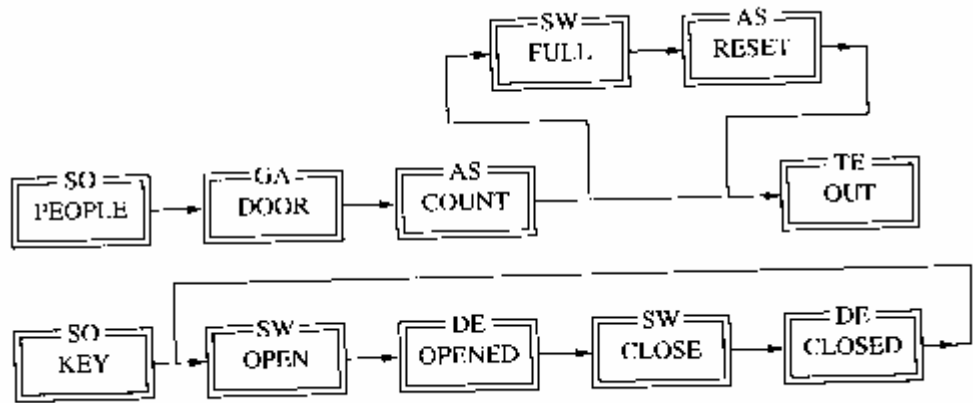


Figura 5.14

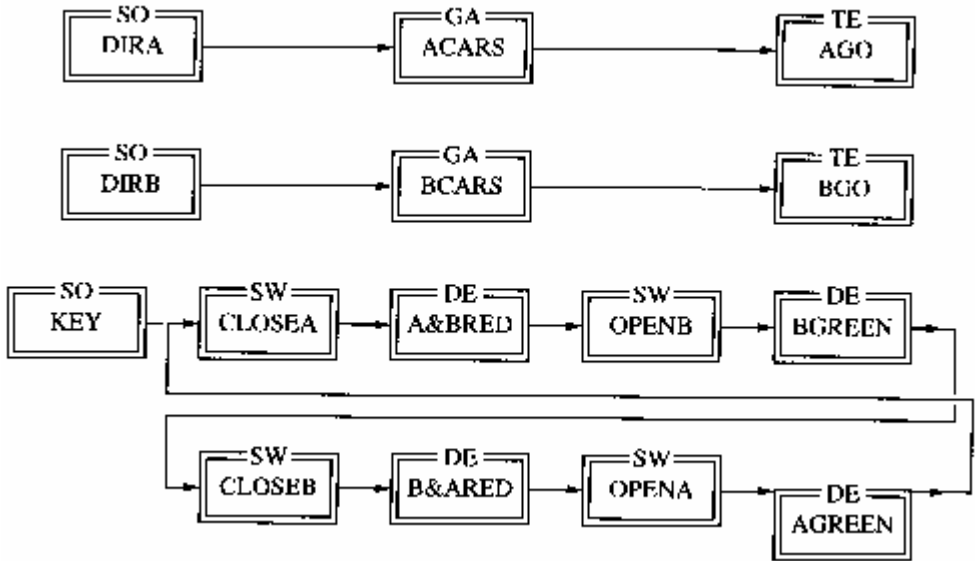
Ilustración de una liberación controlada de personas desde una puerta

Así como las entidades dejan el GATE, entran en el nodo ASSIGN en el cual una variable, digamos COUNTER, es incrementada en uno.  
La rama de este nodo a OUT es del tipo de bifurcación Siempre.  
La rama desde el nodo ASSIGN al nodo SWITCH llamado FULL es condicional y se toma sólo cuando COUNTER llega a 45, indicando que 45 personas han ingresado al showroom.  
En el nodo SWITCH, la puerta está cerrada, y en el subsecuente nodo ASSIGN el valor de COUNT se fija en cero para preparar el conteo del número de personas entrando al próximo show.  
Para recolectar las estadísticas basadas en la observación sobre el número de personas que no pueden entrar en el momento del show, la longitud de la cola en la puerta (una variable del sistema) puede ser observada cada vez que al nodo FULL llega una entidad.  
Estos requisitos pueden ser fijados en el menú de estadística.

**Ejemplo 5.15**

Este ejemplo demuestra la aplicación de puertas e interruptores para determinar el timing deseable para un semáforo.  
Suponga que un túnel estrecho permite el paso de autos en una dirección por vez.  
Los autos arriban desde las direcciones A y B con tiempos exponenciales entre arribos de 20 y 30 segundos, respectivamente.  
Los semáforos en cada una de las dos entradas del túnel controlan el flujo de tránsito.  
La secuencia de señales es verde en dirección A mientras la otra está roja, roja en ambas direcciones para que los autos despejen el túnel (esto toma 2 minutos), y verde en dirección B mientras la otra está roja.  
Deseamos encontrar el mejor timing de los semáforos que resulte en el menor tiempo total de espera de todos los automóviles.  
Para encontrar el mejor timing, deberíamos tratar sistemáticamente varios pares de tiempos de luz verde en las direcciones A y B, y, para cada par seleccionado, deberíamos simular el sistema y observar el tiempo promedio total de espera de los autos en ambas direcciones.  
Esta investigación puede demandar varias pruebas para alcanzar una solución satisfactoria y puede utilizar una técnica disponible, tal como el método de investigación de pauta adaptativa.  
Dado que aquí la intención es demostrar las capacidades de los nodos GATE y SWITCH, utilizaremos el modelo para un solo par de timings de luz verde, el cual es 4 minutos en dirección A y 3 minutos en dirección B.

La red de este ejemplo se muestra en la **Figura 5.15**.



**Figura 5.15**

Uso de los nodos SWITCH y GATE para una simulación de semáforos

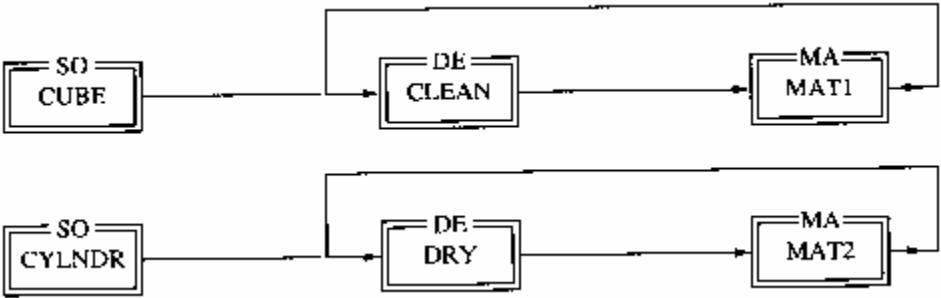
Los dos nodos SOURCE superiores crean los arribos de autos.  
Cada GATE representa la cola de autos delante de la entrada del túnel.  
Ambas puertas están inicialmente cerradas.  
El nodo SOURCE llamado KEY crea una entidad que circula en la red disjunta de abajo y activa los interruptores para repetir la secuencia: abrir la puerta en dirección A, demora por 4 minutos, cerrar la puerta, demora por 2 minutos, abrir la puerta en dirección B, demora de 3 minutos, cerrar la puerta, y demora de 2 minutos.  
Pueden ser simulados varios timings para cada dirección.  
Las estadísticas de las colas para las puertas proveen la información deseada para cada escenario simulado.

**Nodo Coincidir—Q**

El nodo MATCH—Q casa las entidades que arriban a los nodos MATCH asociados.  
Cuando hay una entidad en cada uno de tales nodos con igual valor de un atributo específico, se liberarán todas las entidades para pasar a través de sus correspondientes nodos MATCH simultáneamente.  
Si no se identifican ningún atributo, entonces la primera entidad en cada nodo MATCH—Q será el candidato a casar.  
Un nodo QUEUE no puede preceder a un nodo MATCH—Q porque cada nodo MATCH—Q tiene su propio buffer.  
Todos los nodos MATCH—Q que se suponen casan unos con otros deben tener el mismo nombre alfabético, seguido por un código numérico.  
Por ejemplo, MAT1 y MAT2 indica dos nodos que casan en el grupo de nodos que MAT.  
El proceso de casado tiene parecido con el ensamble con la excepción de que en el ensamble las entidades se unen para formar una entidad, mientras en el proceso de casado cada una de las entidades casadas toma su propia dirección después de haberse casado.

**Ejemplo 5.16**

En un proceso de manufactura, una operación de limpieza es seguida por una de secado.  
Algunos pequeños componentes del producto se ubican en contenedores cúbicos que caben en la máquina de limpieza.  
Después de limpiados, los contenedores se vacían en contenedores cilíndricos que caben en la máquina de secado por centrifugado.  
Hay tres máquinas de limpieza y dos máquinas de secado.  
Cada máquina tiene un contenedor de su tipo.  
Después de haber sido liberados de la correspondiente operación, cada tipo de contenedor espera el arribo del del otro tipo para la transferencia de su contenido.  
Si un contenedor cúbico que ha salido de la máquina de limpieza no encuentra un contenedor cilíndrico para vaciar en él, debe aguardar hasta que uno esté disponible.  
De la misma manera, uno cilíndrico debe aguardar el arribo de uno cúbico.  
El tiempo de limpieza por máquina está distribuido uniformemente entre 2 y 4 minutos.  
El tiempo de secado, por máquina, está distribuido exponencialmente con media 2 minutos.  
Los tiempos da carga y descarga de los contenedores son despreciables.  
Estamos interesados en encontrar el tiempo ocioso promedio de cada clase de contenedor.  
Esto es equivalente al tiempo ocioso promedio de la máquina correspondiente.  
La **Figura 5.16** muestra la red.



**Figura 5.16**

Ilustración de la aplicación del nodo MATCH

Las redes disjuntas superior e inferior en esta figura representan el flujo de los contenedores cúbico y cilíndrico, respectivamente.  
El nodo SOURCE llamado CUBE genera inicialmente tres entidades que representan los contenedores cúbicos.  
El otro nodo SOURCE genera dos entidades que representan los contenedores cilíndricos.  
El tiempo entre estas creaciones es cero.  
Las entidades que representan los contenedores cúbicos se demoran en un nodo DELAY llamado CLEAN y se envían al nodo MATCH—Q para esperar el arribo del contenedor cilíndrico para emparejar.  
Cuando están emparejados un contenedor cilíndrico con uno cúbico, ocurre la transferencia del contenido entre uno y otro, y cada contenedor toma su propia ruta.  
Las estadísticas de la cola para el nodo MATCH—Q provee la información deseada.  
Se recomienda la animación de este modelo para observar la performance del proceso de casado.  
Note que dado que el número de entidades (contenedores) que necesitan las instalaciones (máquinas) es igual al número de instalaciones, las máquinas no se han representado como instalaciones en este ejemplo.

Si se necesitan estadísticas específicas acerca de utilización de máquinas o existe la posibilidad de detenciones, será necesario usar colas y instalaciones con servidores paralelos.

**Nodo Grupo—Q**

El nodo GROUP—Q es utilizado o para combinar o para juntar las entidades que están arribando para formar un único grupo saliente.

Cuando no se pida en un estadio posterior que las entidades agrupadas se desagrupen, las entidades se combinarán para formar una entidad cuyos atributos pueden ser fijados por ciertas reglas, que se proveen en la ventana de selección.

Cuando se pida un posterior desagrupado, las entidades que están arribando se agruparán en un ente saliente en el cual cada entidad preservará sus valores originales de atributos.

Al conjunto saliente se le puede dar un nombre y se le pueden asignar atributos.

Las entidades agrupadas pueden ser desagrupadas más tarde usando un nodo UNGROUP.

Los transportistas pueden ser definidos con varias capacidades y otras características para llevar las entidades agrupadas a través de la red.

Los transportistas son entidades, y cuando se utilizan, el conjunto de atributos de las entidades agrupadas que dejan el nodo GROUP—Q es el conjunto de atributos del transportador que está transportando el producto.

Cuando el transportador arriba a un nodo UNGROUP, puede liberar una porción o todas las entidades agrupadas.

**Ejercicio 5.17**

En una actividad manufacturera, 3 unidades de componente A y 2 unidades de componente B se ensamblan para formar el producto final.

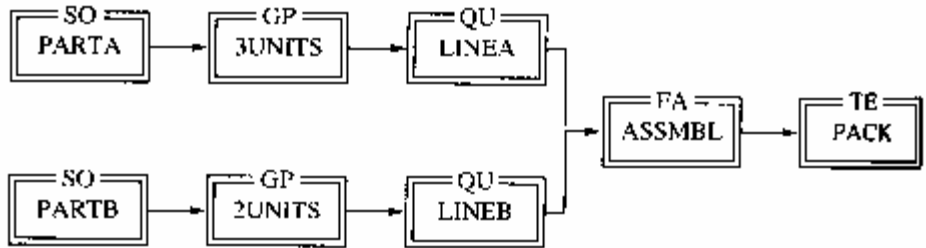
La tasa de arribo promedio de los componentes A y B es conocida.

Los tiempos entre arribos están distribuidos exponencialmente.

La operación de ensamblado toma 30 segundos.

Estamos interesados en encontrar el tamaño del buffer para almacenaje de cada componente antes de la estación de ensamble.

La **Figura 5.17** muestra el diagrama de la red del ejemplo.



**Figura 5.17**

Ilustración de la opción Ensamble en la selección de ramas

Los nodos SOURCE crean los componentes con tiempo entre creaciones distribuido exponencialmente con medias de 20 y 30 segundos para los componentes A y B.

Los nodos GROUP—Q tienen como opción Agrupar por Número, agrupan 3 unidades de componente A en una y 2 unidades de componente B en una, respectivamente.

Las entidades agrupadas van entonces a nodos QUEUE delante del nodo FACILITY.

La opción de selección en el nodo FACILITY es ASM.

Las estadísticas de los nodos GROUP—Q y QUEUE proveen colectivamente la información deseada.

Para observar la perfomance del proceso de agrupado, se recomienda correr el modelo en el modo animación.

**Nodo Desagrupar**

El nodo UNGROUP se utiliza para liberar una porción o todas las entidades agrupadas que arriban a un nodo, con o sin un transportador.

El número posible de ramas emergentes de un nodo UNGROUP es ó 1 ó 2.

Cuando hay sólo una rama emergente, todas las entidades agrupadas se desagrupan y se envían junto con el transportador (si se lo utiliza) a través de la rama emergente.

Cuando hay dos ramas emergentes, una porción o todas las entidades agrupadas se desagrupan y se liberan a través de una rama, mientras que el transportador y las entidades remanentes (si las hay) en el grupo continúan a través de la otra rama emergente.

En el nodo UNGROUP no se permiten más de dos ramas emergentes.

**Ejemplo 5.18**

Los pasajeros llegan a la estación de colectivos de lanzadera de un aeropuerto en el área de estacionamiento de largo plazo con un tiempo entre arribos distribuido exponencialmente con una media de 0,7 minutos.

Hay 5 colectivos de lanzadera operando entre el área de estacionamiento y el aeropuerto.

Cada colectivo transporta un máximo de 20 pasajeros.

El tiempo de viaje de cada colectivo al aeropuerto está distribuido uniformemente en un rango de 7 a 11 minutos.

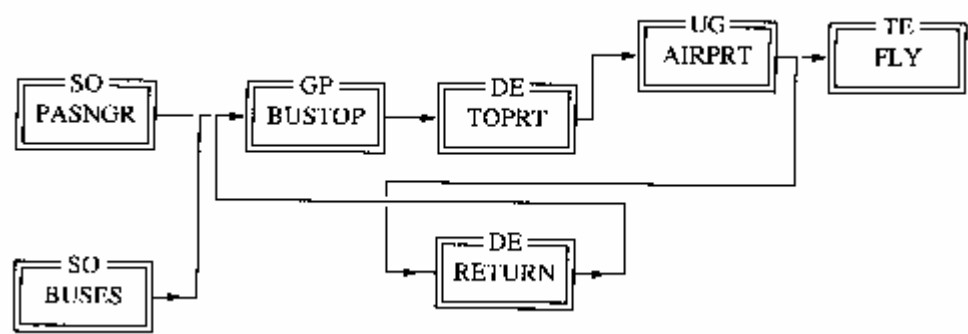
El tiempo de retorno es también uniforme, con un rango de 5 a 9 minutos.

Estamos interesados en las estadísticas de las colas de pasajeros en la estación de lanzadera del área de estacionamiento bajo dos reglas de partida de los colectivos de lanzadera.

Bajo la primera regla operativa, los colectivos toman cualquier número de pasajeros que hubiere en la estación (hasta su capacidad) e inmediatamente salen.

Bajo la segunda regla, los colectivos esperan en la estación hasta que se llenan, partiendo luego hacia el aeropuerto.

La **figura 5.18** muestra la red.



**Figura 5.18**

Ilustración de la aplicación de los nodos GROUP y UNGROUP

El nodo SOURCE llamado PASNGR crea las entidades pasajero que se envían al nodo GRUOP—Q. En este nodo, se define un transportador con el nombre BUS y capacidad 20.

La elección de la regla de agrupamiento en el primer escenario es Disponible, y en el segundo escenario es Capacidad.

EZSIM provee una descripción detallada de los criterios de partida de los transportistas en una ventana de ayuda.

El nodo SOURCE llamado BUSES crea inicialmente cinco colectivos de lanzadera con el nombre de BUS (con un tiempo cero entre creaciones).

Las entidades agrupadas van al nodo DELAY que representa el tiempo de viaje al aeropuerto.

En el nodo UNGROUP llamado AIRPRT, la regla de desagrupar es ALL, que significa que todos los pasajeros salen del colectivo.

La ruta de los pasajeros es al nodo TERMINATE mientras que la ruta de transportistas lleva las entidades BUS de regreso al nodo GROUP a través de un nodo DELAY que representa el tiempo de retorno.

Las estadísticas de la cola proveen la información deseada.

La animación de este modelo provee comprensión de la operación de los nodos GROUP—Q y UNGROUP.

**Nodo Archivo**

El nodo FILE se utiliza para manipular el contenido de las colas en un modelo.

Las operaciones en este nodo incluyen transferir, copiar y eliminar entidades junto con sus atributos.

La posición de una entidad sujeto en una cola (su rango — rank) puede ser definido como último, primero o n—ésimo.

Una porción o todas las entidades (especificadas por un atributo dado) en la cola pueden ser manipuladas simultáneamente por el nodo FILE.

Las operaciones de transferencia y copia pueden ser ejecutadas en el mismo archivo, o sea, las entidades pueden ser transferidas o copiadas de una cola determinada a la misma cola.

Las operaciones de archivo pueden ser ejecutadas como un evento enviando cualquier entidad al nodo FILE en el momento deseado.

**Ejemplo 5.19**

Para una instalación hay dos colas en paralelo.

Cada vez que ocurre una partida, la última entidad en la fila más larga deja la línea para unirse a la otra, si la fila más corta tiene al menos dos entidades en ella.

El tiempo entre arribos para cada línea está distribuido exponencialmente con una media de 10 minutos.

El tiempo de servicio en la instalación está uniformemente distribuido dentro de un rango de 3 a 5 minutos.

La selección de cola en la instalación es al azar.

Estamos interesados en las estadísticas de la cola.

La **Figura 5.19** muestra el diagrama de red de este ejemplo.



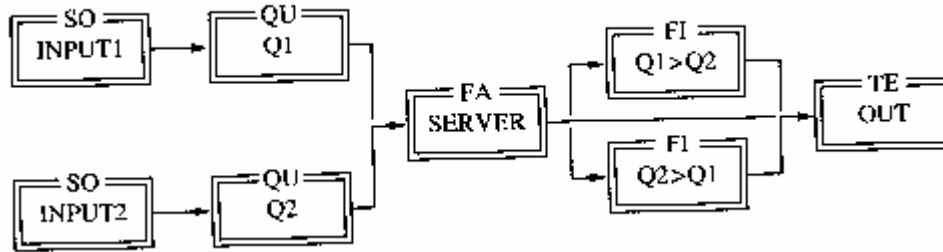


Figura 5.19

Ilustración del uso del nodo FILE para mover entidades entre colas

Las entidades que dejan la instalación toman una de las tres ramas posibles.

Se especifica ramificación condicional para la rama más alta y para la más baja.

La rama más alta va al nodo FILE que toma la última en QUE1 y la ubica en QUE2, si  $LOF(QUE1) > LOF(QUE2) + 1$

La rama de más abajo va a otro nodo FILE, el cual toma la última entidad en QUE2 y la ubica en QUE1, si  $LOF(QUE2) > LOF(QUE1) + 1$

Se puede elegir la opción Expresión para crear las expresiones de las condiciones indicadas más arriba.

La rama del medio va directamente al nodo TERMINATE, y la condición de ramificación Última Opción se especifica para ella.

### Ejercicios

1. En una peluquería trabaja, en un momento dado, sólo un peluquero (por ejemplo, un sistema de un canal). El tiempo entre arribos de los clientes a esta peluquería está distribuido exponencialmente con una media de 12 minutos. El lapso necesario para cortarse el pelo está distribuido normalmente con media de 15 minutos y desvío estándar de 3. Se supone que no existe límite en el número de clientes que pueden estar esperando y que una vez que el cliente llega a la peluquería no partirá hasta que se le haya cortado el pelo. Inicialmente no hay clientes aguardando y el peluquero está ocioso. Haga una simulación de la peluquería por un período de 8 horas. El objetivo de esta simulación es estimar el tiempo medio de espera de cada cliente en la peluquería.
2. En una fábrica, se producen partes a una velocidad de una cada 5 minutos. Una estación de inspección verifica cada parte. El 85 % de las unidades se aceptan y se envían al departamento despacho. Los conjuntos rechazados van a una estación de reparación donde hay dos operadores trabajando en paralelo. Entonces las partes reparadas se envían a la estación de inspección. El tiempo de inspección está distribuido uniformemente entre 3 y 5 minutos. El tiempo de reparación para cada operador está distribuido exponencialmente con una media de 7 minutos. Simule este sistema por 1.000 unidades de tiempo para determinar los tamaños de los buffer necesarios en las áreas de inspección y reparación. ¿Usted recomendaría un operador de inspección y/o reparador adicional? Encuentre las estadísticas del tiempo total que cada parte pasa en el sistema.
3. En el **Ejemplo 2** suponga que aquellas partes que necesitan reparación por segunda vez son desechadas. Modele el sistema y simule el modelo hasta que ocurran tres descartes (puede utilizar un nodo TERMINATE con un conteo de entidades hasta 3 para las desechadas.)
4. En una terminal aérea, los aviones de carga con capacidades de 50 ó 70 unidades de carga arriban a la velocidad de uno cada 60 minutos. Cada avión espera hasta que se carga su capacidad y se va. El 55 % de los aviones que arriban tienen una capacidad de 50 unidades. La carga arriba en grupos de 10 unidades de carga con tiempo entre arribos distribuido exponencialmente con media de 9 minutos. Modele las unidades de carga como recursos y corra el modelo de simulación de este modelo para 100 llegadas de aviones para determinar el tiempo de espera promedio de cada avión antes de su partida.
5. 40 % de los clientes que llegan a una peluquería requieren peinado, el resto corte. El tiempo de peinado es  $Exp(15)$  y el de corte es  $Exp(8)$ . Hay un peluquero en el negocio. El tiempo entre arribos de los clientes es  $Exp(12)$ . Simule este sistema por 1.000 unidades de tiempo, (a) suponiendo al peluquero como una instalación y (b) suponiendo que el peluquero es un recurso. Compare los resultados.



# ANEXO A

Se incluyen algunos ejemplos de interés.

## Ejemplo

### **Ejemplo de sistema estocástico** [Hillier y Lieberman]

Se tiene la oportunidad de participar de un juego en el que se lanza una moneda hasta que la diferencia entre caras y cruces sea 3. Se deben pagar 1 UM pero se reciben 8 UM cuando se gana.

Se gana dinero si el número de lanzamientos necesarios es menor a 8 y se pierde si es mayor a 8. ¿Conviene o no participar?

Podría hacerse tirando una moneda y viendo que pasa. Sería una **SIMULACIÓN** porque **IMITA** sin perder dinero.

Se puede asociar 0,1,2,3 y 4 con cara y 5,6,7,8 y 9 con cruz. Los números se toman de una tabla de números al azar.

Los números generados por una computadora son:

8 1 3 7 2 7 1 6 5 5 7 9 0 0 3 4 3 5 6 8 5  
8 9 4 8 0 4 8 6 5 3 5 9 2 5 7 9 7 2 9 3 9  
8 5 8 9 2 5 7 6 9 7 6 0 7 3 9 8 2 7 1 0 3  
2 6 2 7 1 3 7 0 4 4 1 8 3 2 1 3 9 5 9 0 5  
0 3 8 7 8 9 5 4 0 8 3 8 0 1

Si se denota H por cara y T por cruz, la primera jugada simulada es:

T H H T H T H T T T T y se necesitaron 11 lanzamientos simulados de la moneda.

Las jugadas subsecuentes requieren 5, 5, 9, 7, 7, 5, 3, 17, 5, 5, 3, 9 y 7 lanzamientos, respectivamente.

Se simularon 14 jugadas.

Un estadístico útil es:

$$\text{Promedio de la muestra} = \frac{11 + 5 + 5 + 9 + 7 + \dots + 7}{14} = 7$$

Parecería que debería jugarse esperando ganar 1 UM por jugada, pero **CUIDADO**

La desviación estándar de la muestra es 3,67 de manera que la desviación estándar estimada del promedio de la muestra es:

$$\frac{3,67}{\sqrt{14}} = 0,98$$

Si se quiere mayor precisión se debe aumentar la muestra.

La media verdadera es 9. A la larga se perdería 1 UM por jugada.

En todos los casos, la simulación trata de representar la realidad por medio de un modelo.

Debe utilizarse en situaciones donde es demasiado costoso o complejo experimentar la situación real. En estos casos, pueden probarse los efectos de una decisión en un modelo de simulación antes de implementar la misma.

Así se han simulado gran cantidad de situaciones, incluyendo el flujo de pacientes de una clínica, la operaciones de redes de transporte físico, la organización de materias en universidades, operaciones en fábricas y llegadas y salidas de todo tipo (buques, aviones, estudiantes, donaciones de sangre, etc.).

La simulación es útil para estudiar problemas de colas con patrones de llegada muy complicados. Los problemas simples de colas se resuelven con modelos matemáticos bastante sencillos.

## Ejemplo [Hillier y Lieberman]

### **Simulación por Método de Montecarlo**

Considérese un juego de dados, donde se tiran dos dados. Si el resultado es 7 u 11 en la primera tirada, ganamos de inmediato. Si sacamos 2, 3 ó 12, perdemos de inmediato. Cualquier otro total, esto es, 4,5,6,8,9 ó 10, nos da una segunda oportunidad. En esta parte del juego, continuamos tirando los dados hasta que obtengamos o un 7 o el total obtenido en la primera tirada. Si obtenemos un 7, perdemos. Si el resultado es el mismo total que en la primera jugada, ganamos. Suponiendo que los dados no están cargados, determine en experimento de simulación para determinar el porcentaje de veces que ganamos.

Gottfried sugiere el siguiente pseudocódigo de PASCAL ISO para detallar este juego de dados:

```
PROGRAM dados(input,output);
CONST origen = 12345;
VAR
    cuenta,x :integer;
    n : real;
```

```

    respuesta : char;
    flag : boolean;
PROCEDURE jugar;
VAR tanteo, punto : integer;

    FUNCTION tirada (VAR : integer);
    VAR d1,d2 : integer;

        FUNCTION rand (VAR X : integer) : real;
        VAR y, z : integer;
        BEGIN
            z:= 1*x;
            IF z >= 0 THEN y:=z
            ELSE BEGIN y := z + 32767; y := y + 1 END;
            rand := 0,3051757 -4*y;
            x :=y
        END;
        BEGIN
            d1 := 1 + trunc (6*rand(x));
            d2 := 1 + trunc (6*rand(x));
            tirada := d1 + d2
        END;
        BEGIN
            readln;
            tanteo := tirada (x);
            CASE tanteo OF
                7,11 : BEGIN
                            writeln ( '   GANA Ud. en la primera jugada ');
                            END;
                2,3, 12 : BEGIN
                            writeln ( '   PIERDE Ud. en la primera jugada ')
                            END;
                4,5,6,8,9,10 : BEGIN
                                    punto := tanteo;
                                    REPEAT
                                        write (tanteo);
                                        readln;
                                        tanteo := tirada (x);
                                    UNTIL (tanteo = punto) OR (tanteo = 7)
                                    IF tanteo = punto
                                        THEN BEGINwrite ( 'GANA Ud. al igua-
                                                lar ') END
                                        ELSE BEGIN write ( ' PIERDE Ud. al sacar
                                                7 ')
                                                END
                                    END
            END
        END
    END;

BEGIN { Rutina ejecutiva }
    x := origen;
    flag := true;
    write ( ' Presione RETORNO DEL CARRO para comenzar a jugar ');
    readln;
    WHILE flag DO
        BEGIN
            jugar;
            write ( ' ¿Quiere Ud. jugar de nuevo? (s/n ') ;
            readln (respuesta);
            IF ( respuesta = 'n ') OR (respuesta = 'N ') THEN flag := false
        END
    END.

```

# ANEXO B

## Números al azar

El generador de números al azar más utilizado es el **GENERADOR CONGRUENTE LINEAL**. Con este método se produce una sucesión de enteros  $x_1, x_2, x_3, \dots$ , entre cero y  $m - 1$  de acuerdo con la siguiente fórmula recursiva:

$$x_{i+1} = (a \cdot x_i + c) \text{ modulo } m \quad (i = 0, 1, 2, \dots)$$

Al valor inicial  $x_0$  se le llama semilla,  $a$  es el multiplicador constante,  $c$  el incremento y  $m$  el módulo. El número aleatorio entre 0 y 1 se genera entonces con la ecuación:

$$R_i = \frac{x_i}{m} \quad (i = 1, 2, 3, \dots)$$

Por ejemplo si:  $x_0 = 35$  ;  $a = 13$  ;  $c = 65$  y  $m = 100$

$$x_1 = (a \cdot x_0 + c) \text{ mod } m = (13 \cdot 35 + 65) \text{ mod } 100 = 20$$

$$R_1 = \frac{20}{100} = 0,20$$

$$x_2 = (13 \cdot 20 + 65) \text{ mod } 100 = 325 \text{ mod } 100 = 25$$

## Generación de Números Aleatorios utilizando Excel®

Para generar número aleatorios se pica en Herramientas, Análisis de Datos, Generación de Números Aleatorios

10 números aleatorios generados con Excel.	10 números aleatorios generados con Excel.
Distribución Uniforme	Distribución Normal
72	0,68
6	-0,72
77	-0,15
79	-0,68
86	1,06
95	0,00
13	2,13
60	-0,39
75	-1,38
45	-1,51



# ANEXO C

## SIMULACIÓN CON VARIABLES ALEATORIAS CONTINUAS

### MÉTODO DE LA TRANSFORMACIÓN INVERSA

Se usa, en general, para funciones cuya función de distribución acumulada se puede obtener en forma cerrada, o sea:

- EXPONENCIAL;
- UNIFORME;
- TRIANGULAR;
- WEIBULL.

Se hace en tres pasos:

**PASO 1:**

Dada una función de densidad de probabilidad  $f(x)$  para una variable aleatoria  $x$ , obtener la función de distribución acumulada  $F(x)$  como:

$$F(x) = \int_{-\infty}^x f(t) dt$$

**PASO 2:**

Generar un número aleatorio  $r$ .

**PASO 3:**

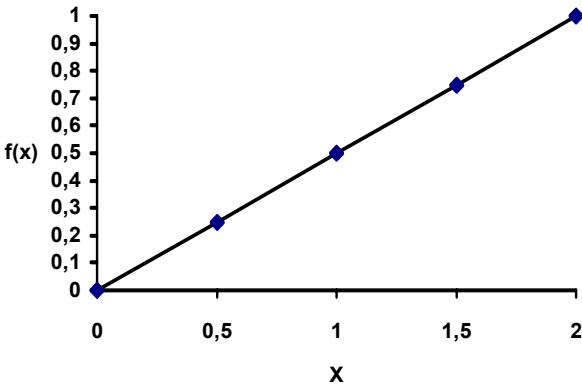
Hacer  $F(x) = r$  y despejar  $x$ . La variable  $x$  es entonces un número aleatorio procedente de la distribución cuya función de densidad de probabilidad es  $f(x)$

### EJEMPLO

Se tiene la función de densidad de probabilidad representada por:

$$f(x) = \begin{cases} \frac{x}{2} & 0 \leq x \leq 2 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Se llama función de rampa.



El área bajo la curva  $f(x) = \frac{1}{2} x$  representa la probabilidad de ocurrencia de la variable aleatoria  $x$ .

Suponemos que en este caso  $x$  representa los tiempos de servicio de un cajero de banco. Para obtener valores aleatorios a partir de esta distribución mediante el método de transformación inversa, calculamos primero la función densidad acumulada como:

$$F(x) = \int_0^x \frac{t}{2} dt = \frac{x^2}{4}$$

Que se representa mediante la función:

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{x^2}{4} & 0 \leq x < 2 \\ 1 & x \geq 2 \end{cases}$$

Con el PASO 2 generamos un número aleatorio  $x$ . Luego, con el PASO 3 hacemos  $F(x) = r$  y calculamos  $x$ .

$$\frac{x^2}{4} = r$$

$$x = \pm 2\sqrt{r}$$

Los tiempos de servicio se definen para valores positivos de  $x$ ; no es posible un valor  $x = -2\sqrt{r}$ . Queda  $x = 2\sqrt{r}$  como solución de  $x$ .

A esta ecuación se la llama GENERADOR DE VALOR ALEATORIO o GENERADOR DE PROCESO.

Así, para obtener un tiempo de servicio generamos primero un número aleatorio y luego lo transformamos por medio de la ecuación anterior.

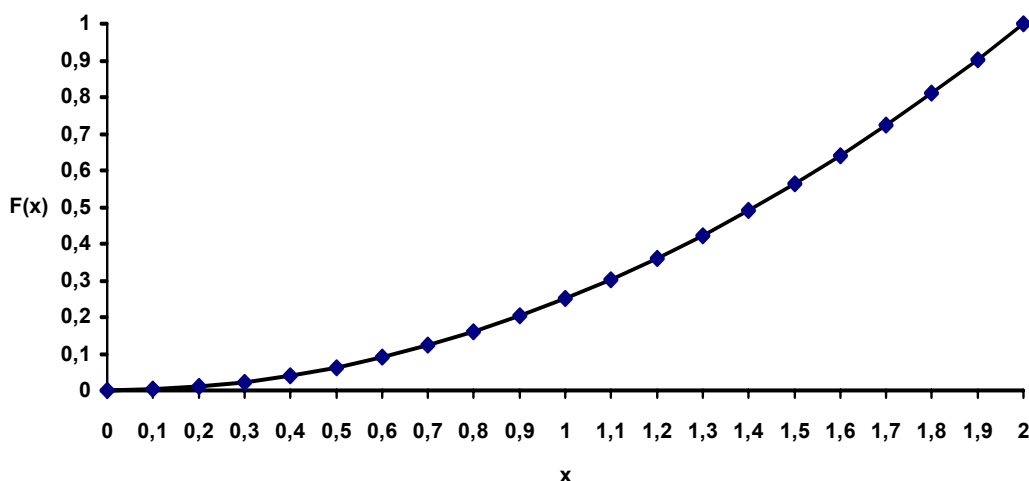
Por ejemplo, si se obtiene un número aleatorio

$$r = 0,64$$

se generará un tiempo de servicio

$$x = 2\sqrt{0,64} = 1,6$$

El método de la transformación inversa se puede graficar:



El intervalo de valores para la variable aleatoria es

$$0 \leq x \leq 2$$

el que coincide con las probabilidades acumuladas

$$0 \leq F(x) \leq 1$$

Para cualquier valor de  $F(x)$  en el intervalo  $[0,1]$  existe un valor correspondiente de la variable aleatoria, representada por  $x$ .

Como un número aleatorio se define en el intervalo entre 0 y 1, esto indica que el número aleatorio se puede traducir en forma directa a un valor correspondiente a  $x$  mediante la relación

$$r = F(x)$$

La solución para  $x$  en términos de  $r$  se conoce como el cálculo de la inversa de  $F(x)$  y se representa por  $x = F^{-1}(r)$

De aquí el nombre de representación inversa. Nótese que si  $r$  es igual a cero, generamos una variable aleatoria igual a cero, el valor más pequeño posible de  $x$ .

Si generamos un número aleatorio igual a 1 se transformará en 2, el valor más grande posible para  $x$ .

Para demostrar que el método de transformación inversa genera números con la misma distribución que  $x$  considere el hecho de que para dos números

$$x_1 \text{ y } x_2$$

cualesquiera la probabilidad

$$P(x_1 \leq x \leq x_2) = F(x_2) - F(x_1)$$

Lo que debemos demostrar es que la probabilidad de que el valor generado de  $x$  esté entre  $x_1$  y  $x_2$  también es la misma.

En el gráfico vemos que el valor generado de  $x$  quedará entre  $x_1$  y  $x_2$  sí y sólo si el número aleatorio escogido queda entre  $r_1 = F(x_1)$  y  $r_2 = F(x_2)$

Así, la probabilidad de que el valor generado de  $x$  esté entre  $x_1$  y  $x_2$  también es  $F(x_2) - F(x_1)$

Esto demuestra que el método de transformación inversa en realidad genera números con la misma distribución que  $x$ .

## LA DISTRIBUCIÓN EXPONENCIAL



La distribución exponencial tiene aplicaciones importantes en la representación matemática de los sistemas de colas de espera.

La exponencial está dada por:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 ; \lambda > 0 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Debemos generar observaciones a partir de la distribución exponencial.

En el PASO 1 determinamos la función de distribución acumulada.

Está dada por:

$$F(x) = \begin{cases} 0 & x < 0 \\ 1 - e^{-\lambda x} & x \geq 0 \end{cases}$$

Hacemos

$$F(x) = r$$

para despejar  $x$ .

$$1 - e^{-\lambda x} = r$$

$$e^{-\lambda x} = 1 - r$$

Sacando logaritmo natural a ambos lados

$$-\lambda x = \ln(1 - r)$$

$$x = -\frac{1}{\lambda} \ln(1 - r)$$

Para simplificar nuestros cálculos, podemos reemplazar

$$(1 - r)$$

por

$$r$$

Como  $r$  es un número aleatorio,

$$(1 - r)$$

también será aleatorio.

Y así nuestro generador será

$$x = -\frac{1}{\lambda} \ln r$$

Por ejemplo, si

$$r = \frac{1}{e} ; x = \frac{1}{\lambda}$$

y

$$r = 1 ; x = 0$$

## DISTRIBUCIÓN UNIFORME

Se tiene una variable aleatoria  $x$  que se distribuye uniformemente en el intervalo  $[a, b]$  La función de densidad de probabilidad para este caso puede ser representada por:

$$f(x) = \begin{cases} \frac{1}{b - a} & a \leq x \leq b \\ 0 & \text{en cualquier otro caso} \end{cases}$$

La función de densidad de probabilidad acumulada para este caso es:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x - a}{b - a} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

Si hacemos  $F(x) = r$

$$\frac{x - a}{b - a} = r$$

De donde:

$$x = a + (b - a)r$$

Si  $r = \frac{1}{2}$

$$x = a + \frac{b - a}{2} = \frac{2a + b - a}{2} = \frac{a + b}{2}$$

Si  $r = 1$

$$x = a + (b - a) = a + b - a = b$$

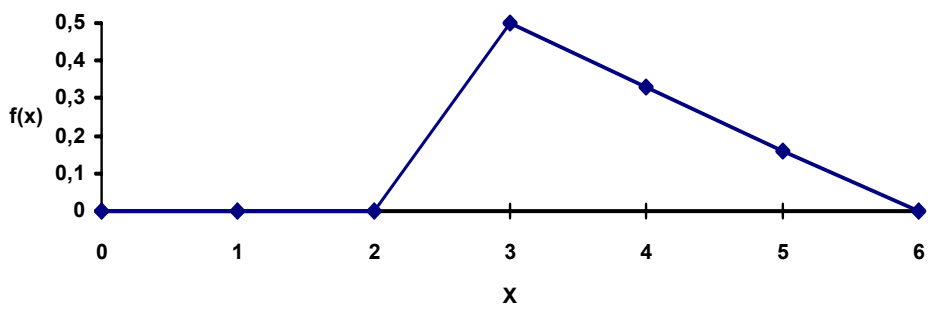
Si  $r = 0$

$$x = a$$

**DISTRIBUCIÓN TRIANGULAR**

Se tiene una variable aleatoria  $x$  cuya función de densidad de probabilidad es:

$$f(x) = \begin{cases} \frac{1}{2}(x - 2) & 2 \leq x \leq 3 \\ \frac{1}{2}\left(2 - \frac{x}{3}\right) & 3 \leq x \leq 6 \\ 0 & \text{en cualquier otro caso} \end{cases}$$



La función de densidad acumulada es:

$$F(x) = \begin{cases} 0 & x < 2 \\ \frac{1}{4}(x - 2)^2 & 2 \leq x \leq 3 \\ -\frac{1}{12}(x^2 - 12x + 24) & 3 \leq x \leq 6 \end{cases}$$

Por simplicidad definiremos a

$$F(x) = \frac{1}{4}(x - 2)^2$$

como

$$F_1(x) \text{ para } 2 \leq x \leq 3$$

y

$$F(x) = -\frac{1}{12}(x^2 - 12x + 24)$$

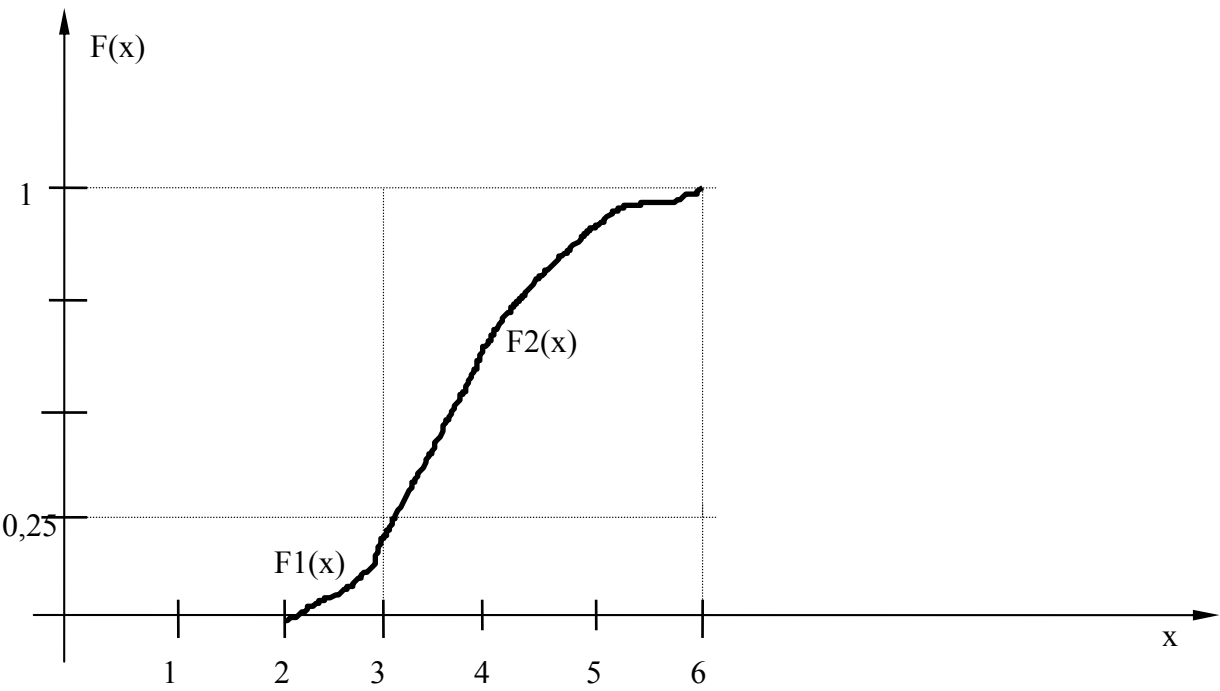
como

$$F_2(x)$$

para

$$3 \leq x \leq 6$$

La función de densidad de probabilidad acumulada es:



Nótese cuando  $x = 3$  ;  $F(3) = 0,25$ .

Esto indica que la función  $F_1(x)$  abarca el primer 25 % del intervalo de la función de densidad de probabilidad acumulada y que  $F_2(x)$  se aplica al 75 % restante del intervalo.

Ahora tenemos dos funciones separadas que representan la función de distribución acumulada.

$$\frac{1}{4}(x-2)^2 = r \text{ para } 0 \leq r < 0,25$$

$$-\frac{1}{12}(x^2 - 12x + 24) = r \text{ para } 0,25 \leq r \leq 1$$

Y  $x$  será la variable aleatoria de interés.

De la primera:

$$x - 2 = \pm\sqrt{4r}$$

$$x = 2 \pm \sqrt{4r}$$

$$x = 2 \pm 2\sqrt{r}$$

$x = 2 - 2\sqrt{r}$  no es posible por lo que:

$$x = 2 + 2\sqrt{r}$$

si  $r = 0$

$$x = 2 + 0 = 2$$

si  $r = 0,25$

$$x = 2 + 2\sqrt{\frac{1}{4}} = 2 + \frac{2}{2} = 2 + 1 = 3$$

Para la segunda ecuación

$$-\frac{1}{12}(x^2 - 2x + 24)$$

$$x = 6 \pm 2\sqrt{3-3r}$$

Como  $x$  debe ser menor a 6 la parte

$$x = 6 + 2\sqrt{3-3r} \text{ no se usa.}$$

Usaremos sólo

$$x = 6 - 2\sqrt{3-3r}$$

Si  $r = 0,25$

$$x = 6 - 2\sqrt{3-3\frac{1}{4}} = 6 - 2\sqrt{3-\frac{3}{4}} = 6 - 2\sqrt{\frac{9}{4}} = 6 - 2\frac{3}{2} = 3$$

Si  $r = 1$

$$x = 6 - 2\sqrt{3-3\cdot 1} = 6$$

## EJEMPLO DE SIMULACIÓN ESTOCÁSTICA

### Ejemplo

Cabot Inc. es una gran firma de pedidos por correo en Chicago.

Los pedidos llegan al almacén vía telefónica. En la actualidad, Cabot mantiene 10 operadoras al teléfono las 24 horas del día.

Las operadoras toman los pedidos y los alimentan directamente a una computadora central, por medio de terminales.

Cada operadora tiene una terminal.

En la actualidad se cuenta con 11 terminales.

O sea, si todas las operadoras están trabajando, habrá una terminal desocupada.

Una terminal en línea puede descomponerse.

En este caso la terminal se quita de la estación de trabajo y se reemplaza por una que no esté ocupada.

Si ninguna está disponible, la operadora deberá esperar a que haya una.

Durante este tiempo la operadora no toma pedidos.

La terminal descompuesta se manda al taller, donde la empresa tiene un canal de atención asignado a reparar terminales.

Cuando se termina de reparar, la terminal queda en reserva o pasa directamente al servicio.

Los gerentes de Cabot creen que se necesita evaluar el sistema de terminales, porque ha sido mucho el tiempo de inactividad de las operadoras a causa de terminales descompuestas.

Sienten que se puede resolver el problema si compran algunas terminales más para que estén en el grupo de reserva.

Los contadores han determinado que una terminal nueva costará 75 UM semanales en costos como inversión, capital, mantenimiento y seguros.

Los costos del tiempo muerto de una terminal se han estimado en 1.000 UM/semana.

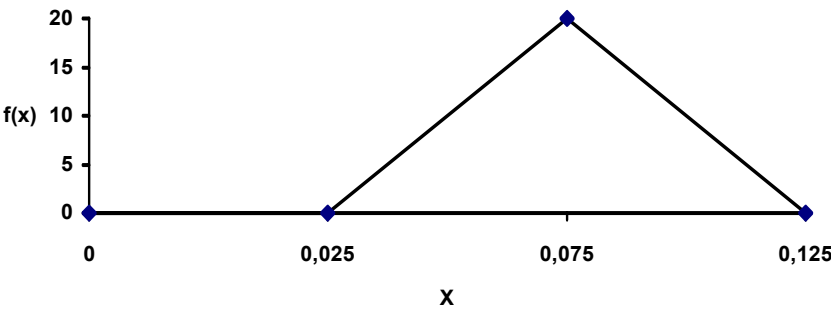
Se desea saber cuántas terminales más se deben comprar.

SOLUCIÓN

Los tiempos de reparación sólo se pueden representar por una distribución triangular.  
Debemos emplear simulación.  
Para la distribución de tiempos muertos los datos demuestran que la distribución es exponencial con media 1 semana por terminal.  
El análisis de los tiempos de reparación, medidos en semanas, se puede representar mediante la distribución triangular:

$$f(x)=\begin{cases} -10+400\ x & 0,025\leq x\leq 0,075 \\ 50-400\ x & 0,075\leq x\leq 0,125 \end{cases}$$

que tiene una media de 0,075 por semana, o sea que el personal puede reparar, en media, 13,33 terminales por semana.  
Para determinar el número óptimo de terminales para el sistema, debemos equilibrar el costo de terminales adicionales y los mayores ingresos que se generan como resultado de un aumento en el número de terminales.



En la simulación, aumentaremos el número de terminales  $n$  en el sistema, desde el actual total de 11, en incrementos de 1.

Para este valor fijo de  $n$  ejecutamos el modelo de simulación para estimar el ingreso neto.  
Seguimos agregando terminales hasta que el ingreso neto alcance su máximo.  
Para calcular el ingreso neto, calcularemos primero el número medio de terminales en funcionamiento,

$$EL_n$$

o, de manera equivalente, el número promedio de terminales que no funcionan,

$$ED_n$$

para un número fijo de terminales en el sistema,  $n$ .

De hecho,

$$ED_n \text{ es igual a } n - EL_n$$

Una vez que se tiene un valor para

$$EL_n$$

podemos calcular los costos semanales esperados de tiempo muerto dados por:

$$1.000(10 - EL_n) \text{ si } EL_n < 10$$

Entonces el aumento en los ingresos, como resultado de aumentar el número de terminales de 11 a  $n$  es igual a:

$$1.000(EL_n - EL_{11})$$

La fórmula matemática para calcular

$$EL_n$$

es:

$$EL_n = \frac{\int_0^T N(t) dt}{T} = \frac{\sum_{i=1}^m A_i}{T}$$

Donde:

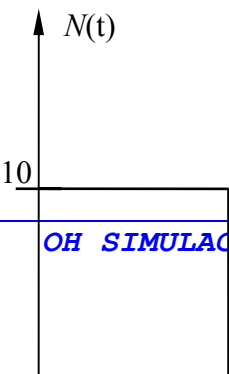
$T$  : longitud de la simulación;

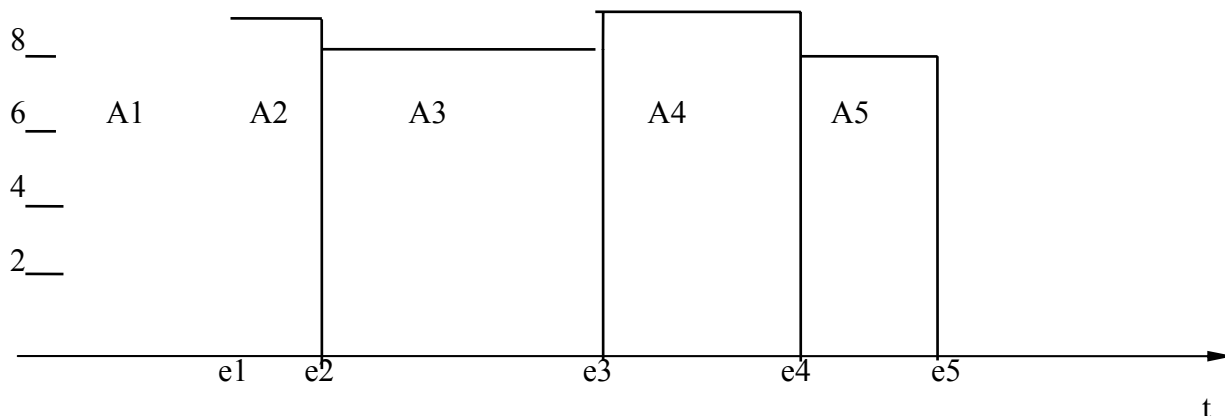
$N(t)$  : número de terminales trabajando cuando el tiempo es  $t$  ( $0 \leq t \leq T$ );

$A_i$  : área del rectángulo bajo  $N(t)$  entre  $e_{i-1}$  y  $e_i$ , donde  $e_i$  es el tiempo del  $i$ -ésimo evento;

$m$  : número de eventos que suceden en el intervalo  $[0, T]$

En el gráfico tenemos  $n = 10$





Iniciamos con 10 unidades en línea cuando el tiempo es cero.

Entre 0 y  $e_1$  el tiempo total para todas las terminales es:

$$10 e_1$$

El tiempo total en línea entre los eventos 1 y 2 es:

$$9(e_2 - e_1)$$

Si ejecutamos esta simulación  $T$  unidades de tiempo y sumamos las áreas  $A_1, A_2, \dots$  podemos obtener una estimación de  $EL_{10}$  al dividir las áreas  $A_1, A_2, \dots$  por  $T$ .

Tenemos una medida estadística llamada PROMEDIO TEMPORAL.

Si se hace la simulación durante un período de tiempo suficientemente largo, la estimación de  $EL_{10}$  debe acercarse mucho al promedio real.

Estado del sistema = número de terminales en el taller de reparación.

Hay dos eventos en la simulación = descompostura y terminación de la reparación.

Las descomposturas se presentan con distribución exponencial; el generador del proceso es:

$$x = -\log r$$

En el caso de los tiempos de reparación, al aplicar el método de transformación inversa tenemos:

$$x = 0,025 + \sqrt{0,005 r} \quad (0 \leq r \leq 0,5)$$

y

$$x = 0,125 - \sqrt{0,005(1-r)} \quad (0,5 \leq r \leq 1,0)$$

en este experimento ejecutamos diversas simulaciones, una para cada  $n$  distinta y aumentamos hasta que los ingresos netos alcancen un máximo.

Generamos una variable aleatoria exponencial para cada terminal en línea a partir de la distribución de descomposturas, y al establecer la hora de descompostura al sumar este tiempo con la hora actual del reloj, que es cero.

Una vez programados estos eventos, determinamos el primero, que es la primera descompostura, al buscar en la lista actual de eventos y procesamos esta descompostura.

Para procesar una descompostura, tomamos dos series separadas:

(1) Determinar si está disponible una reserva. Si hay una disponible, ponerla en servicio y programar su tiempo de descompostura. Si ninguna está disponible, actualizar el estado de pedidos atrasados.

(2) Determinar si el personal de reparación está desocupado. Si es así, iniciar la reparación de la terminal descompuesta mediante la generación de una variable aleatoria de la distribución de tiempos de servicio y programar la hora de terminación de la reparación.

Si el personal de reparación está ocupado, colocar la terminal descompuesta en la cola de reparación.

Habiendo completado estas dos series de acciones, actualizamos los contadores estadísticos.

Para procesar la terminación de una reparación, también se emprenderán dos series de acciones:

(1) Al terminar una reparación tenemos una terminal más en buen estado y, entonces, determinamos si la terminal pasa en forma directa a una operadora o a la reserva. Si existe un pedido atrasado, ponemos la terminal directamente en servicio y programamos la hora de descompostura para esta terminal del modo normal.

Si no hay operadora que espere una terminal, ésta pasa a la reserva.

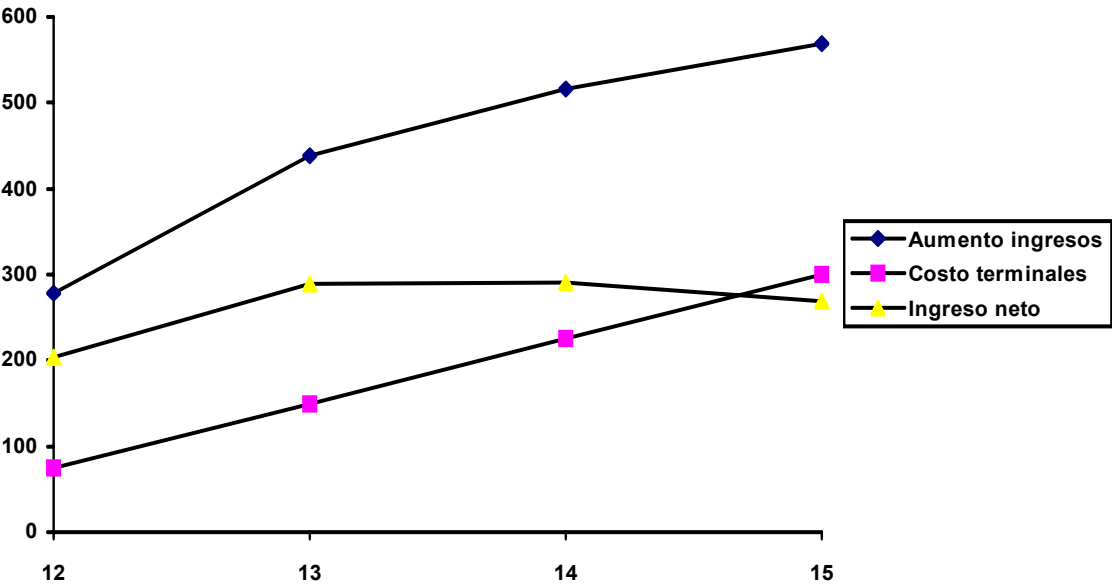
(2) Se verifica la cola de reparación para cerciorarnos si hay terminales para reparar. Si la cola es mayor que cero, llevamos la primera terminal de la cola al taller y programamos la hora de terminación de esa reparación. En cualquier caso ponemos en desocupado el estado del personal de reparación. Por último, al terminar estas acciones actualizamos todos los contadores estadísticos.

Procedemos con la simulación (para una  $n$  dada) al pasar de un evento a otro hasta llegar al tiempo de terminación  $T$ . En este tiempo calculamos todas las medidas relevantes de desempeño a partir de los contadores estadísticos. Nuestra medida clave es el valor actual de  $n$ . Si este ingreso es mayor que el de un sistema con  $n - 1$  terminales, aumentamos en uno el valor de  $n$  y repetimos la simulación con  $n + 1$  terminales en el sistema.

Si no es así, el ingreso neto ha alcanzado un máximo. Si éste es el caso, detenemos el experimento y aceptamos a  $n - 1$  como el número óptimo de terminales que hay que tener en el sistema.

Cuando  $n$  de 11 a 12 el número esperado de terminales en línea aumenta de 9,362 a 9,641, un incremento neto de 0,279. Esto ocasiona un aumento de 279 UM en los ingresos semanales a un costo de 75 UM dándonos un incremento de los ingresos netos de 204 UM por semana. Si aumentamos de 11 a 13 tenemos un aumento neto de 289 UM. El aumento neto llega a un máximo con 14 terminales en el sistema. Si tabulamos y representamos:

	Número de terminales $[n]$				
	11	12	13	14	15
$EL_n$	9,362	9,641	9,801	9,878	9,931
$EL_n - EL_{11}$	—	0,279	0,439	0,516	0,569
Aumento ingresos	—	279	439	516	569
Costo terminales	—	75	150	225	300
Ingreso neto	—	204	289	291	269





ANEXO D

EJEMPLO DE SIMULACIÓN CON EXCEL®  
PIZZERIA MARIO

La Pizzeria de Mario está ubicada cerca del campus de una universidad y provee almuerzos. Mario ha recolectado datos el mes pasado acerca del número de pizzas vendidas durante el período del almuerzo en los días hábiles. Se resumen los datos.

Demanda	Probabilidad
45	0,15
46	0,15
47	0,25
48	0,20
49	0,15
50	0,10

Mario cada mañana recibe 48 medias mazas de pizza (a partir de ahora MMP). Las MMP no utilizadas se pueden guardar para el día siguiente. Se hace la suposición de que Mario comienza la simulación sin inventario de MMP. Vamos a desarrollar una hoja de cálculo para simular 2 semanas de operación (excluyendo fines de semana) para determinar el promedio y el máximo nivel del inventario. También se debe hacer que el modelo calcule el número máximo y el promedio de las rupturas de stock.

Solución

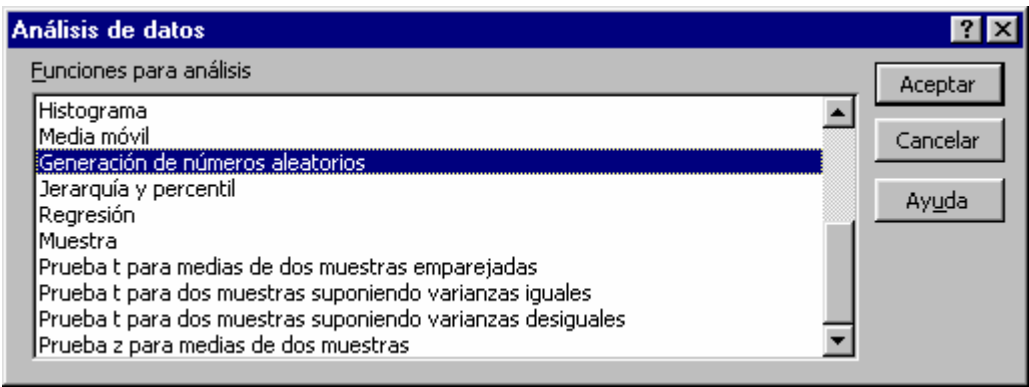
El primer paso para hacer una simulación Montecarlo a mano es definir como son asignados los números aleatorios a los diferentes resultados. Utilizando Excel® sólo se deben especificar la demanda y su probabilidad. Se utilizará la siguiente hoja de cálculo:

Día		Inventario	Demanda	Inventario	Rupturas
Número	Recepción	Inicial	Generada	Final	Inventario
1	48	0	47	1	0
2	48	1	45	4	0
3	48	4	48	4	0
4	48	4	49	3	0
5	48	3	49	2	0
6	48	2	50	0	0
7	48	0	45	3	0
8	48	3	47	4	0
9	48	4	49	3	0
10	48	3	45	6	0
Promedio				3	0
Máximo				6	0

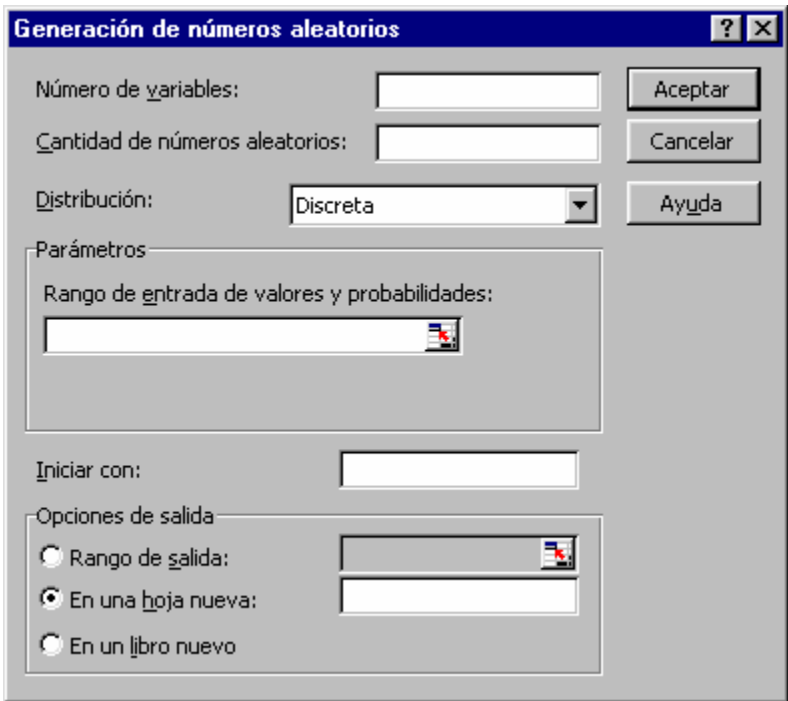
Demand a	Probabilidad
45	0,15
46	0,15
47	0,25
48	0,20
49	0,15
50	0,10

La columna A se utiliza para indicar el día. La columna B contiene las recepciones diarias de MMP. En general, Mario ordena y recibe 48 MMP cada mañana. La columna C contiene el inventario inicial de MMP de cada día. Se comienza la simulación sin inventario de MMP, por lo que se ingresa 0 en la celda C3. Dado que el inventario final del día 1 se convierte en el inicial del día 2, se ingresa =E3 en la celda C4. Dado que esta lógica se aplica en todo el lapso de 2 semanas de simulación del modelo, la fórmula en C4 puede ser copiada a las celdas C5:C12. La columna D contiene las demandas generadas al azar basadas en la distribución de probabilidad ingresada en las celdas H6:I11. Se utiliza la característica Generación de Números al Azar de Excel. Para hacerlo se pica en Herramientas, luego en Análisis de Datos.

Aparece el cuadro de diálogo de análisis de datos:



Se selecciona Generación de Números al Azar y aparece el cuadro de diálogo:



En nuestro caso hay una sola variable (demanda) para la que necesitamos generar números aleatorios. Por lo tanto en la caja "Número de variables" ingrese el número 1.

Luego, dado que necesitamos generar 10 números aleatorios (uno para cada uno de los diez días que estamos simulando, o las celdas D3:D12) ingrese 10 en la caja "Cantidad de números aleatorios".

El tipo de distribución que tenemos es discreta, por lo que dejamos la caja "Distribución" sin cambios. Nótese que Excel puede generar números aleatorios para varias distribuciones, incluyendo uniforme, normal, poisson y binomial.

Para la caja "Rango de entrada de valores y probabilidades", especificamos el rango que contiene los valores de demanda y las probabilidades, que están contenidos en las celdas H6:I11.

Se usa un valor semilla para iniciar la simulación.

La ventaja de especificar un valor semilla es que cada vez que se corre la simulación utilizando una semilla en particular, se genera la misma secuencia de números al azar.

O sea que especificando un valor semilla permite repetir la simulación con la misma secuencia de números al azar.

También se utiliza para reducir la variación al simular diferentes guiones.

O sea, si se utilizan diferentes semillas para verificar políticas diferentes, quien toma la decisión puede no saber si los resultados dependen de diferencias reales en los números al azar utilizados o en diferencias reales entre las políticas.

Para especificar un valor semilla se lo puede ingresar en la caja "Iniciar con"

Si no se especifica nada, lo elige Excel.

Para este ejemplo, dejaremos que Excel elija el valor semilla.

Finalmente, en la sección Opciones de Salida, seleccione el botón "Rango de salida" e ingrese el rango D3:D12 en la caja correspondiente para decirle a Excel dónde colocar los valores generados.

El cuadro de diálogo Generación de Números al Azar queda como se muestra:

Seleccione Aceptar en esta ventana para generar un conjunto de 10 números basados en la distribución ingresada en las celdas H6:I11 para ser ubicadas en las celdas D3:D12.

En la columna E se calcula el inventario al finalizar el día.

En la columna F se calcula el número de pizzas que fueron demandadas pero que no pudieron suplirse dado que no había stock suficiente.

Nótese que para un día dado sólo una de estas cantidades puede ser positiva.

O sea, no es posible tener inventario sobrante y ruptura de stock el mismo día.

La lógica para calcular el inventario sobrante es:

1. Si la suma de recepción más inventario de inicio es mayor que la demanda del día, el inventario final es igual a lo recibido más inventario inicial menos demanda.
2. De lo contrario el inventario final es cero.

Para capturar la lógica se utiliza la función IF.

La función IF tiene tres argumentos separados por puntos y comas, con la siguiente sintaxis

=IF(condición;valor retornado si condición es verdadero;valor retornado si condición es falso)

Para calcular el inventario final del primer día, se ingresa la siguiente fórmula en la celda C3:

=IF(B3+C3>D3;B3+C3—D3;0)

La fórmula puede ser copiada ahora a las celdas E4:E12

Para calcular la ruptura de inventario del primer día se utiliza:

=IF(D3>B3+C3;D3—B3—C3;0)

Esta fórmula puede copiarse a las celdas F4:F12

Las mediciones de perfomance para el modelo de simulación se ingresan en las celdas E13:F14.

La celda E13 calcula el inventario sobrante promedio de los 10 días simulados con la fórmula =PROMEDIO(E3:E12).

En forma similar el máximo se calcula en la celda E14 con la fórmula =MAX(E3:E12).

Se copia luego a F13:F14.

El valor real de los modelos de simulación no es utilizarlos una vez sino muchas para ver cómo se desempeña bajo una gran variedad de situaciones

Es muy fácil en una hoja de cálculo.

Pique Herramientas, Análisis de datos, Generación de números al Azar y Aceptar. Se genera una nueva corrida.



Actualizado al 7/1/2.000

📁 D:\INVESTIGACIÓN OPERATIVA\OH SIMULACIÓN