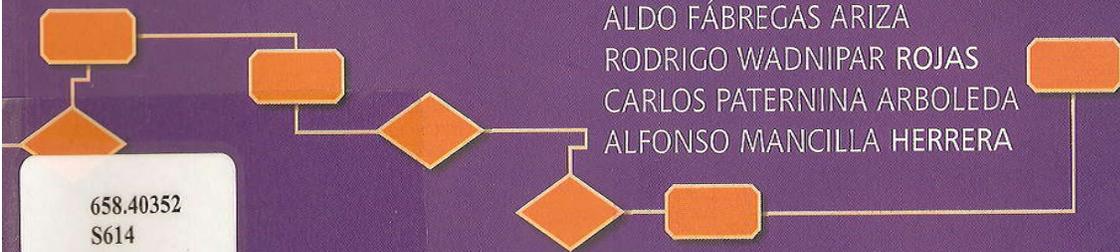


Ediciones Uninorte



SIMULACIÓN DE
SISTEMAS
PRODUCTIVOS CON

ARENA[®]



ALDO FÁBREGAS ARIZA
RODRIGO WADNIPAR ROJAS
CARLOS PATERNINA ARBOLEDA
ALFONSO MANCILLA HERRERA

658.40352
S614
Ej.1

SIMULACIÓN DE SISTEMAS
PRODUCTIVOS CON
ARENA[®]

Aldo Fábregas Ariza
Rodrigo Wadnibar Rojas
Carlos Paternina Arboleda
Alfonso Mancilla Herrera

Ediciones Uninorte
Barranquilla, Colombia



574.191
V172

Fábregas, Aldo, et al.

Simulación de sistemas productivos con RS Arena® /
Aldo Fábregas, et al. — Barranquilla : Ediciones Uninorte,
2003

208 p.

ISBN: 958-8133-24-6

1. SIMULACIÓN
2. OPTIMIZACIÓN
- i. Tit.



ACREDITADA INSTITUCIONALMENTE
POR SU EXCELENCIA ACADÉMICA

Resolución del MEN N° 2085-05/09/2003

- © Ediciones Uninorte, 2003
© Aldo Fábregas Ariza, Rodrigo Wadnibar Rojas,
Carlos Paternina Arboleda, Alfonso Mancilla Herrera, 2003

Coordinación editorial
Zoila Sotomayor O.

Diseño y diagramación
Luz Miriam Giraldo Mejía

Corrección de textos
Mariela González Hawkins

Diseño de portada
Joaquín Camargo Valle

UNIVERSIDAD DEL MAGDALENA BIBLIOTECA CENTRAL		
No.	EL _____	
Fecha:	Septiembre 4 2003	
Valor: \$	_____	
Librería:	Opinión nuevas tecnologías	
Compra ()	Comie ()	Donación (x)

Impreso y hecho en Colombia
Cargraphics S.A.
Santafé de Bogotá
Printed and made in Colombia

CONTENIDO

Prefacio	1
1	
CONCEPTOS BÁSICOS	3
Fundamentos de Arena [®] , 3. Fundamentos de la simulación, 5.	
2	
PANEL DE PROCESOS BÁSICOS.....	9
Módulos lógicos, 9. Módulos de datos, 10. Manejo del flujo de entidades, atributos y variables, 17.	
3	
TRATAMIENTO DE COSTOS	39
Costo inicial, 40. Costo de permanencia, 40. Costo de procesamiento, 41	
4	
ANIMACIÓN	51
Animación de entidades, 51. Cambio en la animación de entidades,54. Animación de un recurso, 55. Animación del valor de una variable en el modelo, 56. Gráficas, 62.	
5	
GENERACIÓN Y ANÁLISIS DE REPORTES	65
Control de corridas, 65. Manejo de reportes, 67. Interpolación de resultados, 71.	
6	
PANEL DE PROCESOS AVANZADOS	75
Módulos lógicos, 75. Variaciones en la capacidad de los recursos, 77. Empleo de entidades de control, 83. Cómo buscar y remover entidades específicos de una cola, 91. Modelo de ensamble, 97. Cómo guardar estadísticas en un archivo externo, 100. Cómo recoger un grupo de entidades en una cola y depositarlo en otra parte del modelo, 105. Alternativas de uso, 109.	

7

ANALIZADOR DE DATOS DE ENTRADA 111

Entrada al *input analyzer*, 111. Preparación de los archivos de datos, 112. Creación de un archivo de entrada, 115. Ajuste de una distribución específica de datos de entrada, 116. Generación de datos, 119. Opciones del histograma, 120.

8

PANEL DE TRANSFERENCIA AVANZADA 123

Funcionamiento de un transporte, 123. Solicitud de transporte, 124. Animación, 133. Comida y análisis de resultados, 136.

9

HERRAMIENTAS DE ANÁLISIS 139

Process analyzer, 139. Recolección de estadísticas, 155. *Output analyzer*, 166.

10

INTEFARZ ARENA®-EXCEL®. 178

Creación del modelo Arena®, 177. Creación de la hoja de cálculo en Excel, 180. Ejecución de la Interfaz, 188.

ANEXO 193

Distribuciones estadísticas

PREFACIO

En el ejercicio de las funciones típicas de cualquier actividad humana, el hombre tiene que tomar decisiones de uno y otro tipo en forma permanente. Esta situación trae consigo, por lo general, riesgo e incertidumbre, lo que compromete la calidad y el logro de la decisión; esto lo dificulta el hecho de que el funcionario responsable debe enfrentar la presión que implica la alta responsabilidad involucrada y, en ciertos casos, su inexperiencia o incompetencia. Para contrarrestar esta situación, el hombre ha desarrollado a través del tiempo una diversidad de herramientas que le permiten minimizar el riesgo y la incertidumbre en la toma de decisiones.

La simulación es una de estas herramientas; con su aplicación no sólo se logra el anterior cometido, sino que se minimizan los costos involucrados en la decisión mediante un mejor uso de los recursos, la disminución del tiempo utilizado y la minimización de las probabilidades de riesgo. A través del proceso de diseño de un modelo de un sistema real, y dirigiendo el experimento con él, se puede entender el comportamiento del sistema, lo cual permite tomar mejores decisiones.

De acuerdo con el diccionario *Webster*, la simulación pretende obtener la esencia de algo sin contar con la realidad. Según *Schriber* (1987), implica la modelación de procesos o sistemas, de tal manera que el modelo emula la respuesta de los sistemas actuales o propuestos como eventos que tienen lugar en el tiempo.

La aplicación de la simulación para buscar la esencia de un sistema implica, por lo general, el manejo de un volumen considerable de datos y la ejecución de un alto número de repeticiones del proceso, ya que se pretende lograr una adecuada historia artificial que permita tomar una decisión con un alto grado de confiabilidad; sólo es factible que este manejo se haga en el computador con la ayuda de un *software* especializado; de este tipo existen en el mercado una amplia gama, como ARENA[®], *software* que se estudia en este documento. La ausencia de una guía práctica para el aprendizaje y manejo de ARENA[®] versión 5.0, motivó la elaboración de este libro, el cual tiene como objetivo constituirse en un manual de consulta y referencia para los usuarios o para quienes desean aprender a utilizar este *software*.

Rodrigo Wadnipar Rojas

CONCEPTOS BÁSICOS

1. FUNDAMENTOS DE ARENA®

1.1. RESEÑA HISTÓRICA

ARENA tiene sus orígenes en 1982; en ese año Dennis Pegden publicó el primer lenguaje de simulación de propósito general para modelar sistemas de manufactura en un PC. Esta aplicación constaba de varias características de manufactura de carácter especial, que hacían que el lenguaje fuera bastante útil y eficiente en el momento de modelar sistemas grandes y complejos.

Pero fue sólo en 1993 cuando se introdujo el sistema de modelamiento ARENA®, el cual crea ambientes gráficos e interactivos para el diseño de modelos mediante el uso del lenguaje SIMAN. Con base en este lenguaje se pueden crear modelos que simulen áreas específicas de los procesos, como el transporte de elementos y la comunicación, entre muchos otros.

1.2. ¿QUÉ ES ARENA®?

Es un sistema que provee un entorno de trabajo integrado para construir modelos de simulación en una amplia variedad de campos; integra, en un ambiente fácilmente comprensible, todas las funciones necesarias para el desarrollo de una simulación exitosa (animaciones, análisis de entrada y salida de datos y verificación del modelo).

El desarrollo de modelos de simulación mediante este sistema tiene varias ventajas, entre las que se encuentran:

- Es una poderosa herramienta de simulación.
- Comprende un entorno amigable, que está especialmente diseñado para personas que no poseen conocimientos de programación.
- Los utilitarios que brinda son de fácil uso.
- Cuenta con una excelente capacidad gráfica.
- Ofrece gran versatilidad, pues se puede modelar desde una fábrica automotriz hasta una sala de espera de un hospital.
- Es compatible con productos MICROSOFT OFFICE®.

Sin embargo, ARENA[®] tiene también algunos puntos débiles, a saber:

- Es difícil correr un modelo creado en ARENA[®] en cualquier otro programa de simulación. Esto se da básicamente porque es difícil sincronizar los relojes con los que funcionan los programas.
- La edición para estudiantes tiene muchas limitaciones.
- La documentación y la ayuda que este sistema ofrece es poca; además, no es lo suficientemente clara, y algunos de los ejemplos que presenta tienen errores.

1.3. COMPATIBILIDAD CON MICROSOFT OFFICE[®]

Anteriormente se señaló que la compatibilidad con los productos MICROSOFT OFFICE[®] es una de las características principales de ARENA[®]; esto quiere decir, entre otros aspectos, que las barras de herramientas, los menús y hasta las teclas aceleradoras de este sistema son similares a las que usa MICROSOFT OFFICE[®]. Así, quienes hayan tenido algún tipo de relación con estos productos se darán cuenta que muchos de los procesos que se realizan en OFFICE se ejecutan de forma análoga en ARENA[®].

Entre los aspectos comunes de ambos sistemas se encuentran:

Menús estándar. Los menús de ARENA[®] siguen los mismos lineamientos en la organización de los menús de las aplicaciones OFFICE. Es por ello que ARENA[®] también cuenta con las opciones *File* (Archivo), *Edit* (Edición), *View* (Ver), estándares *Window* (Ventana) y *Help* (ayuda), ampliamente usadas en los productos OFFICE.

Barras de herramientas y Tool Tips. ARENA[®] cuenta con unas barras de herramientas similares a las que poseen las aplicaciones OFFICE. Por ejemplo, se puede imprimir un documento simplemente presionando el botón *Print* en la barra de herramientas estándar; además, se puede conocer la función de los botones al posicionarse sobre cada uno de ellos; cuando se ejecuta este proceso, el *Tool Tip* despliega por pantalla el nombre del botón. Así mismo, es posible reorganizar las barras de herramientas mediante el uso del arrastre y pegue, que es una de las principales características de los productos OFFICE.

La integración con OFFICE permite, además, que se acceda a todas las funciones de ARENA[®] desde cualquier programa de la familia MICROSOFT. Ésta es una característica de gran importancia, ya que los resultados que se obtienen de la simulación, y con el fin de ejecutar su manipulación, se pueden llevar en forma inmediata y directa a una hoja de cálculo o a un procesador de texto.

Mediante OFFICE se pueden crear, diseñar, ejecutar, modificar y examinar modelos de ARENA[®]; es decir, en cualquier momento, y haciendo uso de los macros de los productos OFFICE, se puede abrir y crear un modelo de ARENA[®] y controlar su ejecución. La comunicación entre ambos sistemas se realiza mediante los objetos DAO que utilizan funciones de transferencia de datos entre módulos. Este tema se desarrolla en el capítulo 10.

2. FUNDAMENTOS DE LA SIMULACIÓN

2.1 ENTIDAD

Puede ser un objeto o persona que se mueve a través de un sistema y que causa cambios en las variables de respuesta.

Ejemplos de entidades y sistemas:

- Un cliente en un banco
- Una orden de pedido en un sistema de inventarios
- Una lámina de acero en un proceso de manufactura

2.2 RECURSO

Es un elemento estacionario que puede ser ocupado por una entidad. Los recursos se emplean cuando se requiere representar actividades claves del sistema que restringen el flujo de entidades.

Los recursos tienen una capacidad finita; así mismo, cuentan con una serie de estados por los cuales atraviesan a lo largo de la simulación, ejemplo, *ocupado*, *ocioso*, *inactivo* o *dañado*. Un recurso puede ser una persona (cajero), una máquina (torno) o, incluso, un espacio en áreas de almacenamiento (zona de carga).

Funcionamiento

Una entidad que solicita un recurso:

- Toma control (*Seize*) del recurso si éste está disponible.
- Si no está disponible, espera en la cola asociada al recurso (*Queue*) hasta cuando éste se desocupe para que tome el respectivo control (*Seize*).

Una entidad que tiene control de un recurso:

- En caso de que no requiera más la atención del recurso, puede liberarlo (*Release*) para darle paso a otra entidad en espera.
- Puede continuar efectuando los procesos que sean necesarios (incluso, en otros recursos) hasta terminar su ciclo en ese recurso y así liberarlo (*Release*).

2.3 ATRIBUTO

Es una característica propia de cada entidad. En ARENA® se pueden definir tantos atributos como lo requiera el usuario para el modelamiento del sistema en estudio. Cada entidad individual tiene su propio valor de atributo. Esto implica que para determinar este valor, a diferencia de las variables, se debe examinar la entidad que lo porta.

Los atributos se definen con un nombre, por ejemplo, *peso*, *número de orden*, *color*, etc., y deben tener un valor numérico que indique algo para el usuario.

Ejemplo: El atributo *color* puede adoptar valores de 1, 2, 3 cuando los colores son amarillo, azul y rojo, respectivamente.

Ejemplo: Se pueden presentar atributos cuyo valor sea único para cada entidad en el sistema, así: número de identificación o código de factura. El número de identificación comienza en 0001 para la entidad 1 y, por cada entidad que ingrese, se incrementa y se hace que este atributo contenga valores únicos.

Los atributos se pueden asignar según la distribución; por ejemplo, en un proceso de envasado de aceitunas, el producto resultante sale con un peso medio que sigue una distribución normal con media de 150.47 gr y una desviación estándar de 5.127 gr. En este caso el programa asigna, para cada entidad y en un atributo llamado *peso*, un valor numérico que sigue esta distribución.

2.4 VARIABLES

Representan características del sistema; son de carácter global, es decir, su valor es el mismo en cualquier parte del modelo. Las variables pueden ser predeterminadas por el programa o definidas por el usuario; se definen con un nombre —por ejemplo, *pedido*— y con un valor numérico que simbolice un estado del sistema.

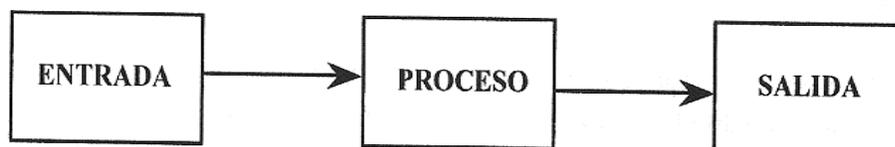
Ejemplo: El número de clientes en una cola puede adoptar valores de 0, 1, 2, 3, etc.; cada valor representa una estado del sistema.

Ejemplo: El nivel de inventario en un sistema logístico puede adoptar un valor que esté por encima del nivel de reorden o que sea menor o igual a este nivel; de esta manera se tienen sólo dos estados de interés representados por múltiples valores de las variables.

2.5 SISTEMA

Es un conjunto de elementos que se encuentran en interacción y que buscan alguna meta o metas comunes; para ello operan sobre dato o información, sobre energía, materia u organismos, con el propósito de producir como salida información, energía, materia u organismos. Un sistema es un conjunto de componentes interrelacionados que, en una forma organizada, recibe entradas y las procesa y emite salidas para obtener una meta común.

Representación de un sistema



Clases de sistemas

- Naturales y artificiales
- Determinísticos y probabilísticos
- Sociales, hombre-máquina y mecánicos
- Abiertos y cerrados
- Permanentes y temporales
- Estables y no estables
- Adaptativos y no adaptativos

2.6 MODELO

Es una representación de la realidad que se desarrolla con el propósito de estudiarla. En la mayoría de los análisis no es necesario considerar todos los detalles; de tal manera, el modelo no sólo es un sustituto de la realidad, sino también una simplificación de ella.

Los modelos se pueden clasificar en:

- Modelos icónicos
- Modelos analógicos
- Modelos simbólicos; estos, a su vez, incluyen:
 - Modelos determinísticos
 - Modelos estocásticos o probabilísticos
 - Modelos dinámicos
 - Modelos estáticos
 - Modelos continuos
 - Modelos discretos

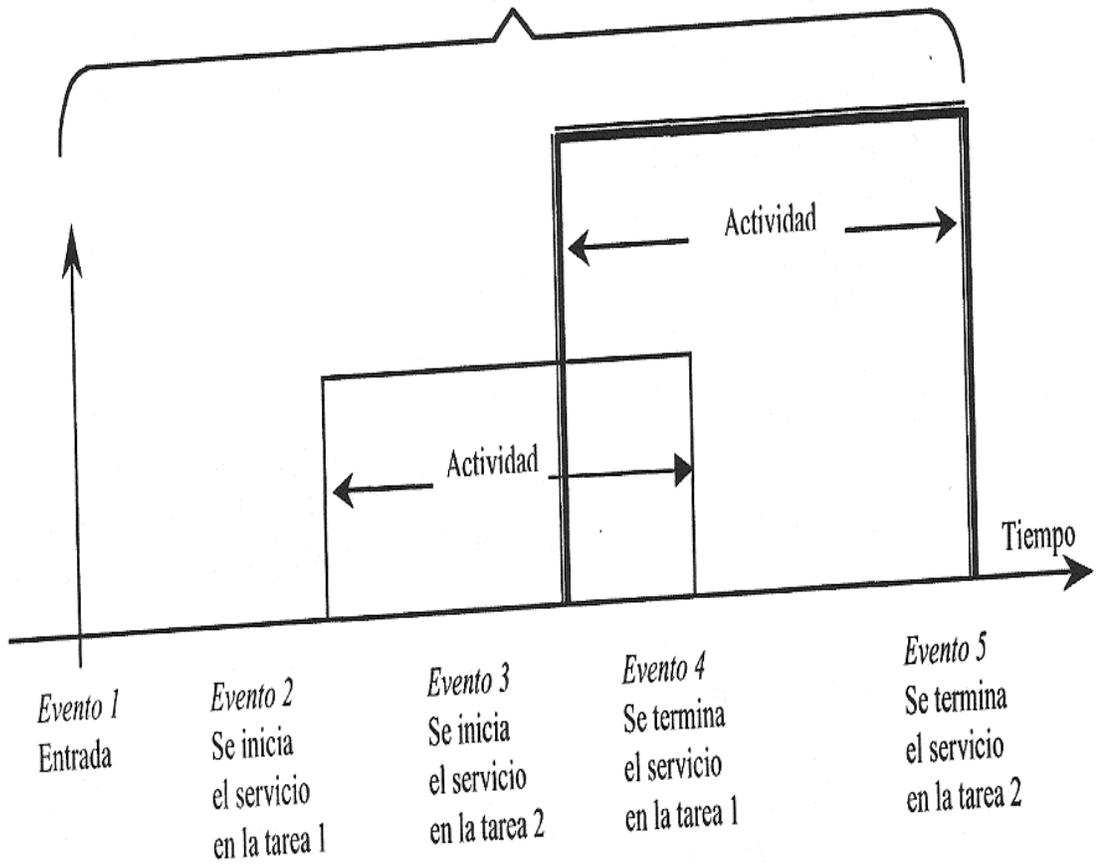
Los modelos tienen las siguientes características:

- Confiabilidad
- Sencillez
- Bajo costo de desarrollo y operación
- Manejabilidad
- Fácil entendimiento del modelo y de los resultados
- La relación costo-beneficio debe ser positiva

2.7 EVENTO

Es la ocurrencia que cambia el estado del sistema. Hay eventos internos y externos; éstos se conocen también como endógenos y exógenos, respectivamente.

Evento, proceso y actividad



PANEL DE PROCESOS BÁSICOS

El panel de procesos básicos (*Basic process panel*) es un conjunto de módulos que permite la elaboración de diagramas de flujo de poca complejidad. Los paneles de modelamiento están divididos en módulos lógicos y módulos de datos. Los primeros hacen parte activa del modelo, ya que modifican atributos o el estado del sistema durante el paso de entidades a través de ellos. Los módulos de datos no hacen parte de la ventana del modelo; vienen en forma de hojas de trabajo, y su función es declarar valores iniciales y propiedades de los diferentes elementos que hacen parte del modelo.

A continuación se reseñan los módulos de ambas categorías con una breve descripción de su función, que se explica con más detalle en instancias posteriores de este capítulo.

1. MÓDULOS LÓGICOS

Create. Las entidades que pasarán a ser procesadas en el sistema simulado se generan en este módulo, el cual se toma como punto de partida para la construcción de un modelo, aunque, según el sistema que se va a simular, varias entradas de entidades pueden ser requeridas. En este módulo se asigna el atributo *Entity Type*.

Dispose. Este módulo, que tiene como función retirar una entidad del modelo, se agrega al final de todas las instrucciones, en el momento en que se considere que la entidad ha llegado al punto de salida del sistema simulado. En este módulo se recolectan las estadísticas correspondientes a la entidad.

Process. En este módulo las entidades experimentan una operación que involucra la utilización de un recurso, la demora que ocasiona el tiempo de procesamiento y la liberación del recurso. Así mismo, en él se puede especificar a qué categoría del costo pertenece el tiempo de la operación (valor agregado, no valor agregado, transferencia, espera, entre otros). Este módulo también puede cumplir la función de un submodelo.

Decide. Permite direccionar el flujo de entidades de acuerdo con una regla de decisión, la cual se puede basar en una condición, en una probabilidad o en una expresión.

Batch. Permite formar lotes o grupos de entidades de un tamaño cualquiera y previamente definido. Estos lotes o grupos pueden ser permanentes o temporales, de cualquier tipo de entidad o de un tipo específico. Cuando se forma un lote se crea una entidad que representa al grupo formado.

Separate. Separa lotes que se forman temporalmente; desarrolla la función opuesta del módulo *Batch*. Este módulo se puede usar también para hacer varias copias de una entidad.

Assign. Su función es cambiar el valor de un atributo, figura, nivel, secuencia u otra variable del sistema. Es posible hacer varias asignaciones en un solo módulo *Assign*.

Record. Se emplea para recolectar estadísticas en el modelo de simulación. También se puede emplear como un contador.

2. MÓDULOS DE DATOS

Entity. En esta hoja de trabajo se definen, en las diferentes categorías, el atributo *Entity Type*, la primera animación que se le asigna a la entidad y sus respectivos costos iniciales.

Queue. En este módulo se definen los nombres de las diferentes colas y el tipo de regla de liberación que éstas siguen (FIFO, LIFO, etc.).

Resource. Los recursos utilizados se declaran en este módulo; así mismo, se definen sus características, como capacidad, costo de operación y fallas.

Variables. En esta hoja se definen los valores iniciales de las variables empleadas en el modelo y, en el caso de los arreglos, sus dimensiones.

Schedule. En este módulo se define el horario de trabajo mediante el cual se programa la capacidad de un recurso.

Sets. Se usa cuando se requiere formar grupos repetitivos de recursos, figuras, colas, contadores, estadísticas, etc., con el fin de facilitar el modelamiento de un sistema determinado.

Ejemplo 2.1

Para ilustrar el funcionamiento del *software* se tomará como punto de partida el caso elemental de *entrada-procesamiento-salida*.

❖ Descripción

Considere el caso de que clientes (entidades) llegan a un cajero (recurso) en un banco con un tiempo entre llegadas que obedece a una distribución normal con media de 5.89 minutos y a una desviación estándar de 0.65 minutos. Los clientes son atendidos por el cajero en un tiempo exponencialmente distribuido con media de 4.5 minutos. Después de ser atendidos, los clientes se retiran del sistema (banco). Simular el sistema por 480 minutos.

■ Desarrollo

Primero se abre una nueva ventana de modelamiento.

a. Creación de entidades

La creación de entidades proporciona el punto inicial del modelo; para ello se agrega un módulo *Create*, lo cual se logra al arrastrar éste hasta la ventana de trabajo.



Create

Luego se procede a editar el módulo. Se hace doble clic en éste para editarlo vía cuadro de diálogo, o se introducen los datos en la hoja de trabajo correspondiente.

Sugerencia. Se pueden editar módulos lógicos (*Create*, *Process*, etc.) y módulos de datos (*Sets*, *Resource*, *Schedule*, etc.) vía cuadro de diálogo, haciendo clic derecho en la primera fila de la hoja de trabajo correspondiente (si no existe fila, debe adicionarse haciendo doble clic en el lugar indicado).

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	Resource 1	Fixed Capacity	1	0.0	0.0	0.0		0 rows

Context menu options:

- Insert Row
- Delete Row
- Edit via Dialog..
- Properties...

• Cuadro de diálogo del módulo Create

Create

Name: Llegada de clientes Entity Type: cliente

Time Between Arrivals

Type: Expression Expression: NORM(5.89,0.65 Units: Minutes

Entities per Arrival: 1 Max Arrivals: Infinite First Creation: 0.0

OK Cancel Help

- Hoja de trabajo del módulo Create

	Name	Entity Type	Type	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Llegada	cliente	Expression	NORM(5.89,0.65)	Minutes	1	Infinite	0.0

Campo	Nombre	Función	Ejemplo
Name	Nombre del módulo	Se utiliza para identificar el módulo y referenciarlo en otras instancias del modelo. Debe ser único.	Llegadas de clientes
Entity Type	Tipo de entidad	Asigna automáticamente el atributo <i>Entity Type</i> , el cual sirve para definir animación y costos iniciales.	Clientes
Type	Tipo de llegada	Determina la forma cómo llegan las entidades al modelo. Puede ser aleatorio (<i>Random</i>), por programación (<i>Schedule</i>), constante (<i>Constant</i>) o una distribución (<i>Expression</i>).	Expression
Entities per arrival	Entidades por evento de llegada	Define el número de entidades que se generan en un evento de llegada; ejemplo, de una en una o de dos en dos, etc. También puede ser una distribución.	1
Max arrivals	Máximo de llegadas	Define el número máximo de llegadas. Cuando se alcanza este campo no se crean más entidades.	Infinite
First creation	Primera creación	Tiempo en el cual se crea la primera entidad. Por defecto es 0.	0

- Tiempo entre llegadas

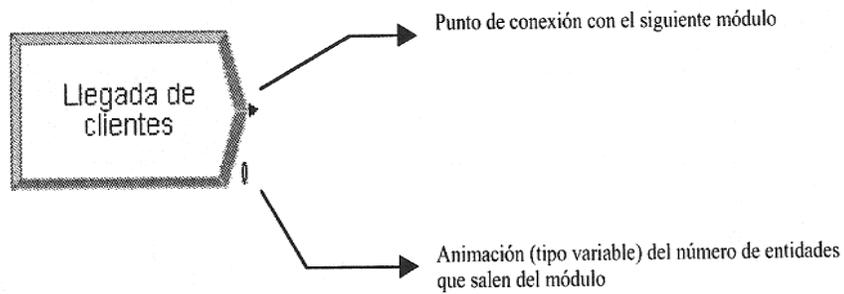
Se considera el tipo, valor y unidad de tiempo; esta última se puede expresar en segundos, minutos, horas o días.

- Hoja de trabajo del módulo Expression

Name	Entity Type	Type	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
Llegada	cliente	Expression	1	Minutes	1	Infinite	0.0
		<ul style="list-style-type: none"> Random Schedule Constant Expression 					

Descripción	Campos	
	Expression	Units
El tiempo entre llegadas corresponde a una distribución o a una variable.	Este campo se llena con la distribución deseada. Las expresiones para las distribuciones vienen expresadas en el menú desplegable asociado.	Unidades de tiempo de llegada. Pueden ser segundos, minutos, horas y días.
Ejemplo	NORM (5.89,0.65)	Minutos

Quando se concluye la edición, el módulo se ve de la siguiente manera:



A continuación se reseñan otros tipos de llegadas:

- *Random*

Descripción	Campos	
	Value	Units
El tipo <i>Random</i> se refiere a una distribución exponencial.	El campo <i>Value</i> se llena con el valor de la media de la distribución.	Unidades de tiempo de llegada. Pueden ser segundos, minutos, horas y días.

- *Schedule*

Descripción	Campos
	Schedule Name
El tiempo de llegadas varía de acuerdo con una programación establecida.	Nombre del elemento <i>Schedule</i> que contiene la programación de llegadas.

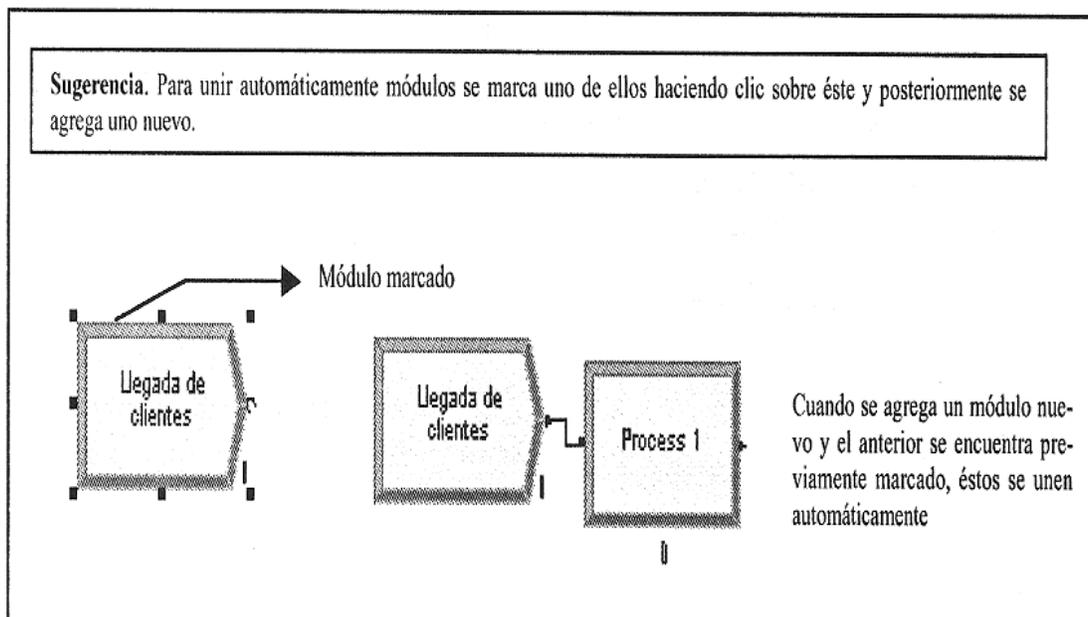
- *Constant*

Descripción	Campos	
	Value	Units
El tiempo entre llegadas es un tiempo constante.	Se llena con el tiempo entre llegadas. Este valor es un número.	Unidades de tiempo de llegada. Pueden ser segundos, minutos, horas o días.

b. Modelamiento de recursos

Para definir una operación o proceso que se le practique a la entidad en un recurso y que tenga una duración asociada, se puede emplear el módulo *Process* si no se requiere de mayor detalle en el modelamiento de esta operación.

El paso que se sigue es agregar un módulo *Process* a continuación del módulo *Create*, y unirlos mediante el conector de tiempo 0 (*Connect*) .



Luego se procede a editar el módulo; para ello se hace doble clic en éste vía cuadro de diálogo o se introducen los datos directamente en la hoja de trabajo correspondiente.

- Cuadro de diálogo del módulo Process

The screenshot shows a dialog box titled "Process". It has several sections:

- Name:** cajeros
- Type:** Standard
- Logic:**
 - Action:** Seize Delay Release
 - Priority:** Medium(2)
- Resources:**
 - Resource, cajero, 1
 - <End of list>
 - Buttons: Add..., Edit..., Delete
- Delay Type:** Expression
- Units:** Minutes
- Allocation:** Value Added
- Expression:** EXPO(4.5)
- Buttons: OK, Cancel, Help

- Hoja de trabajo del módulo Process

	Name	Type	Action	Priority	Resource	Delay Type	Units	Allocation	Expression
1	cajeros	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	EXPO(4.5

Campo	Nombre	Función	Ejemplo
Name	Nombre del módulo	Sirve para identificar el módulo y referenciarlo en otras instancias del modelo. Este campo debe ser único.	Cajeros
Type	Tipo de módulo Process	Define el tipo de módulo. Puede ser estándar (<i>Standard</i>) o submodelo (<i>Submodel</i>).	Standard

Dentro del cuadro de diálogo del módulo *Process* se encuentra un macro denominado *Logic*; en éste se define la acción o acciones que va a seguir la entidad cuando ingrese al módulo en referencia. En el siguiente cuadro se listan y describen las acciones que se pueden llevar a cabo en un módulo *Process*:

Acción	Función
<i>Delay</i>	Durante el tiempo especificado la entidad demora sin ocupar recursos.
<i>Seize, Delay</i>	La entidad ocupa un recurso y sufre una demora. No libera el recurso.
<i>Seize, Delay, Release</i>	La entidad ocupa un recurso, experimenta una demora y, al final, lo libera.
<i>Delay, Release</i>	Una entidad que previamente ha tomado control de un recurso, demora y luego libera a este recurso.
Ejemplo	<i>Seize, Delay, Release.</i>

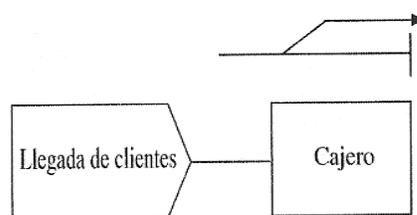
El campo *Priority* se emplea cuando varias entidades se encuentran solicitando la atención de un recurso específico en diferentes puntos del modelo. Las entidades con valores de prioridad bajos se atienden primero que aquellas que tienen valores altos.

A continuación se muestra un macro que contiene los datos de la demora que experimenta la entidad (*Delay Type*). Éste ofrece las posibilidades de uso de una expresión o un tiempo constante, así como se explica en la sección correspondiente a la edición del módulo *Create*. Adicionalmente, ofrece directamente las opciones de una distribución normal, uniforme o triangular, aunque éstas se encuentran contenidas dentro del menú desplegable de expresiones (*Expressions*).

Campo	Ejemplo
<i>Delay Type</i>	Expression
<i>Units</i>	Minutes
<i>Expression</i>	EXPO (4.5)

El campo *Allocate* se utiliza para el tratamiento de costos —este campo se explica en la sección correspondiente al manejo de costos (capítulo 3)—. Si no se están evaluando costos en el modelo, se deja simplemente el valor por defecto de este campo, el cual es *Value-added*

Cuando finaliza la edición, el módulo se ve de la siguiente manera:

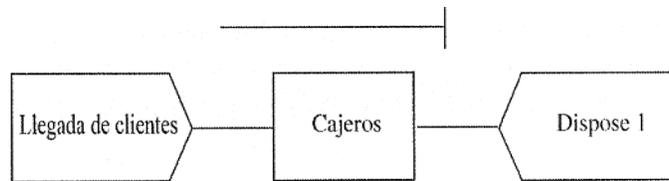


Cuando se define la acción de tomar control de un recurso en el módulo *Process* (*Seize*), automáticamente se crea una cola. Ésta se asocia con el módulo como una de sus propiedades (cajeros. *Queue*), y en ella se almacenarán las entidades que esperan para ser procesadas por el recurso especificado

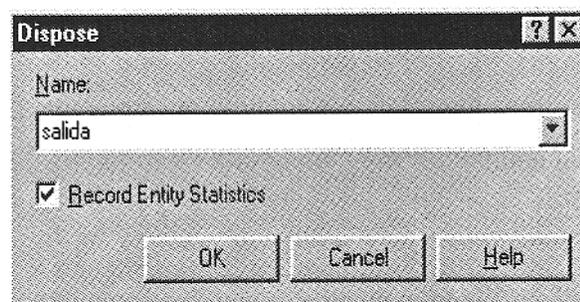
c. Salida de entidades del modelo

Una vez el cajero atiende a un cliente, éste queda desocupado y listo para el siguiente, mientras que la persona que fue atendida se retira del banco (sistema).

Luego se agrega el módulo *Dispose* para retirar entidades del sistema y recolectar estadísticas.



La edición del módulo *Dispose* es bastante simple; sólo se requiere incluir el nombre y activar el cuadro de verificación para la recolección de estadísticas relacionadas con la entidad (*Record Entity Statistics*). Se recomienda la siguiente acción.



3. MANEJO DEL FLUJO DE ENTIDADES, ATRIBUTOS Y VARIABLES

3.1 CREACIÓN DE VARIOS TIPOS DE ENTIDADES

En ocasiones se hace necesario modelar situaciones en las que se tienen varias entidades en un proceso, cada una de las cuales con características definidas que determinan la pertenencia a un grupo específico.

Considere los siguientes ejemplos:

- a. Dos tipos de clientes entran a un banco. Unos van a realizar retiros y otros van a efectuar pagos de cuentas.
- b. Tres tipos de clientes llegan a una estación de servicio automotriz. Algunos llegan a comprar gasolina; otros llevan su auto a mantenimiento, y el resto va a comprar repuestos.
- c. En una compañía productora de lapiceros, antes del ensamble, las diferentes partes son entidades diferentes: tapas, minas y cuerpo.

Para diferenciar las entidades se emplean los atributos. Cuando se le asigna un atributo específico a cierta proporción de entidades que ingresan al sistema, se está creando un grupo especial de éstas. El atributo debe llevar un nombre y tener un valor numérico, así como se explica en la sección *Conceptos básicos* (capítulo 1).

Los atributos pueden ser definidos por el usuario; en algunos casos se pueden utilizar los predeterminados por el programa. En el caso de ARENA[®] versión 8, las entidades cuentan con uno predeterminado llamado *Entity Type*, cuyo valor puede ser modificado

por el usuario cuantas veces considere necesario. Este atributo es asignado a las entidades entrantes en el módulo *Create*, pero se puede cambiar posteriormente. El valor de este atributo debe ser alfanumérico o una mezcla de caracteres permitidos (por ejemplo, tipo1, #1, @1, etc.).

El módulo *Assign* se emplea para la creación o modificación de atributos.



Assign

Ejemplo 2.2

■ Descripción

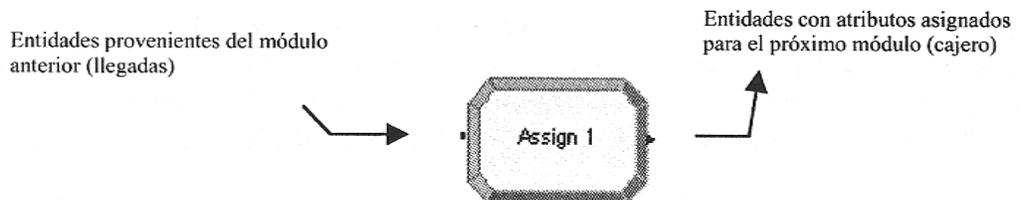
Suponga que se desea modelar la situación que se presenta en el ejemplo 2.1 en las siguientes condiciones:

- Todas las personas tienen el mismo tiempo entre llegadas.
- El 40% de los clientes que llegan al banco va a realizar retiros y el resto va a efectuar pagos de cuentas.
- El atributo que representa el tipo de transacción toma el valor *tran1* (valor alfanumérico) para retiros y el valor *tran 2* para pagos. Este atributo de transacción será el nombre que se le dé al atributo predeterminado *Entity.Type*.
- Existe un solo cajero (recurso) para la atención de clientes.

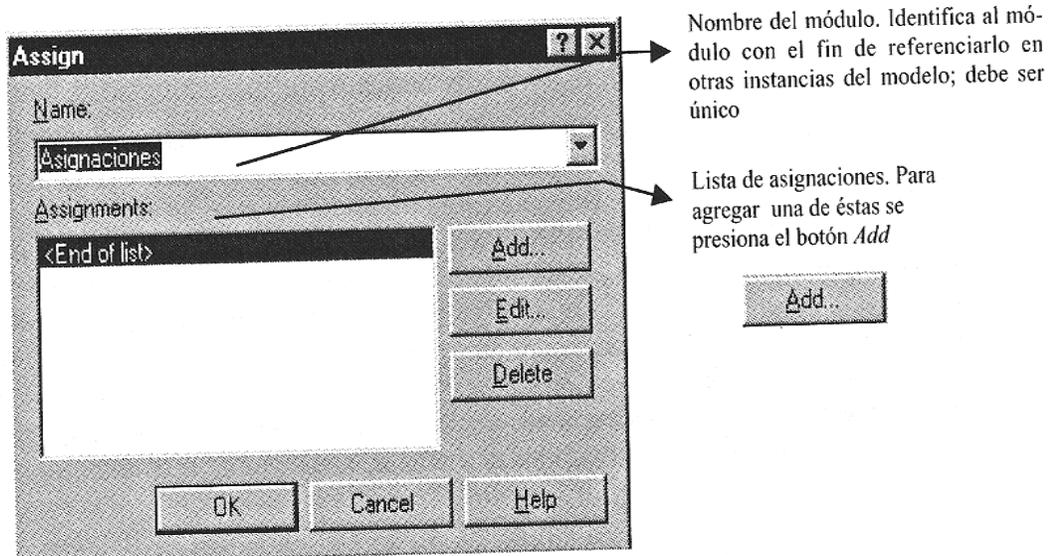
El atributo *Entity.Type* resulta de gran utilidad cuando se diferencian varios tipos de entidades, ya que para cada valor que se especifica de este atributo el programa recolecta estadísticas de manera automática, las cuales se muestran en los reportes correspondientes a las entidades (*Entities*).

■ Desarrollo

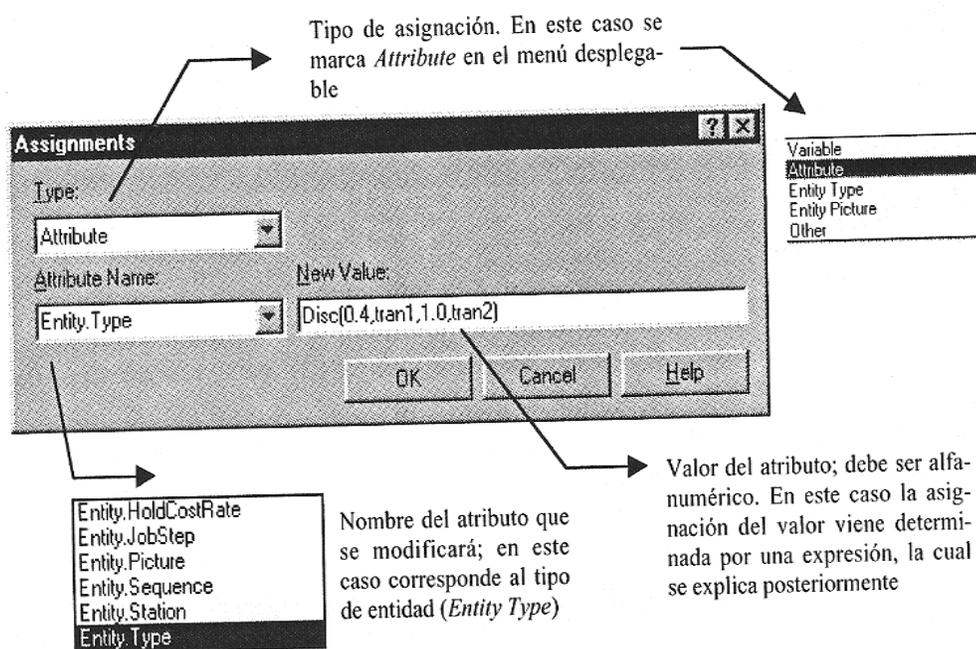
A continuación se explican las funciones del módulo *Assign* para el ejemplo en estudio. Este módulo se debe insertar en el ejemplo 2.1 para su mejor comprensión. En este caso, el valor inicial para el atributo *Entity Type* es *clientes*; en el ejemplo 2.2 el valor se cambia por los nombres o valores antes especificados (*tran1* y *tran 2*).



Inicialmente se hace doble clic sobre el módulo para editarlo vía cuadro de diálogo.



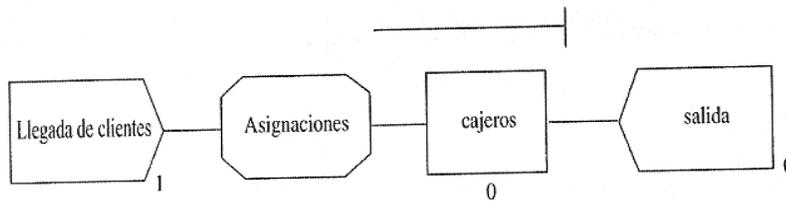
Posteriormente se abre un formulario de asignación:



La expresión utilizada, $DISC(0.4, tran1, 1.0, tran2)$ ¹, representa la asignación del valor *tran1* al 40% de las llegadas y del valor *tran2* al resto para el atributo *Entity Type* (método montecarlo).

¹ Ver: Animación del valor de una variable en el modelo a partir del uso del asistente para construcción de expresiones en el capítulo 4.

Cuando se concluye, el modelo debe quedar de la siguiente forma:



Con este procedimiento se han creado dos tipos de entidades a partir de una sola llegada. Cuando el atributo *Entity Type* es asignado en campos específicos para este propósito (ejemplo, módulo *Create*, campo *Entity Type*), el tipo de entidad correspondiente es definido automáticamente. Cuando se asignan valores para este atributo de manera indirecta, es decir, mediante una expresión —como en el caso actual—, se deben definir los nuevos tipos de entidades; de lo contrario, causará un error en tiempo de corrida.

Con el propósito de definir los tipos de entidades *tran 1* y *tran 2*, se recurre a la hoja de datos de la entidad haciendo clic en el módulo de datos *Entity* en la barra de proyecto.



Entity

Tipo de entidad que se crea automáticamente con el campo del módulo

	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waring Cost
1	cliente	Picture.Report	0.0	0.0	0.0	0.0
2	tran1	Picture.Report	0.0	0.0	0.0	0.0
3	tran2	Picture.Report	0.0	0.0	0.0	0.0

Double-click here to add a new row.

Tipos de entidades definidas por el usuario y que se ingresan manualmente

En el caso de que cada tipo de cliente tenga su propia distribución de llegadas, se hace necesario un módulo *Create* para cada uno.

3.2 DIVISIÓN DEL FLUJO DE ENTIDADES CON BASE EN EL TIPO DE ENTIDAD

Cuando se trabaja en un modelo que contiene varios tipos de entidades, se pueden presentar situaciones en las cuales se necesite separar las que circulan en el sistema de acuerdo con un determinado criterio. Uno de los criterios más útiles son los atributos —como se presenta en la sección *Conceptos básicos* (capítulo 1), éstos son características propias de cada entidad—. Así mismo, el flujo se puede dividir según el estado del sistema o de una condición específica, como se estudia en otras instancias de este manual.

Para seleccionar entidades con base en una condición se emplea el módulo *Decide*.



Decide

Ejemplo 2.3

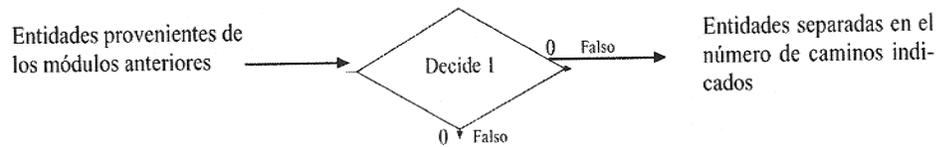
En este caso se muestra la edición del módulo *Decide* dividiendo el flujo de entidades con base en atributos. Este módulo se debe insertar en el ejemplo 2.2 para su mejor comprensión.

■ Descripción

Suponga que se desea contar separadamente con dos tipos de clientes con base en el tipo de transacción que representa el atributo *Entity Type*.

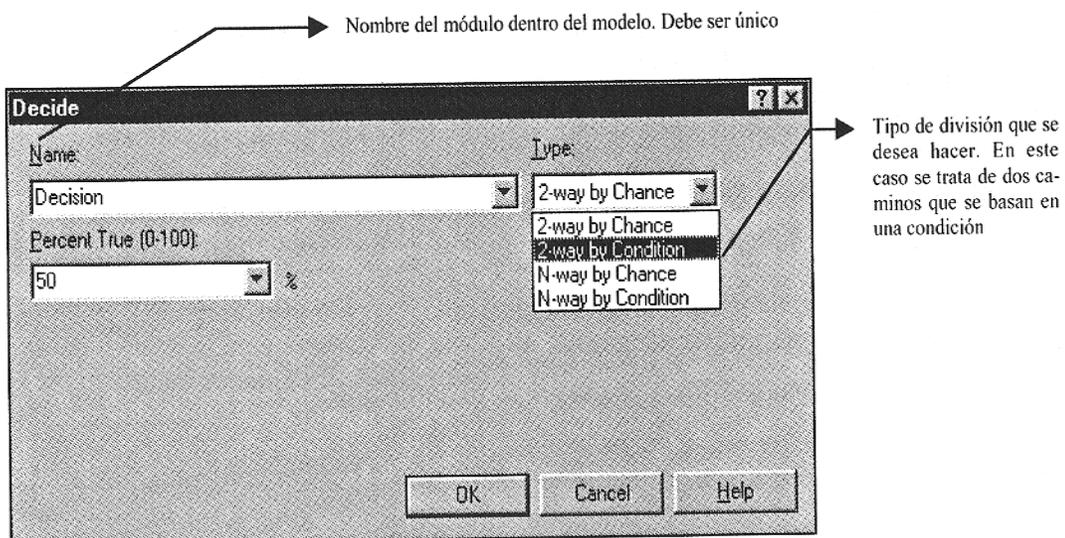
■ Desarrollo

Inicialmente se agrega un módulo *Decide*, así como se muestra a continuación:



Operación	Valor del atributo <i>Entity Type</i>
Retiros	Tran 1
Pagos	Tran 2

Después se hace doble clic sobre el módulo, con el propósito de editarlo vía cuadro de diálogo.



El cuadro de diálogo cambia, y aparecen campos para llenar; éstos obedecen a una sentencia lógica *if-then*, que asigna una instrucción con base en la verificación de una condición. Es decir, si *Entity.Type* es igual a *tran 1*, se cumple la condición y la salida del bloque *Decide* es por la rama *true* (verdadero).

Si el tipo de entidad
(*Entity.Type*)

Llamado *tran1* (retiros)

If:

Entity Type
Variable
Attribute
Entity.Type
Expression

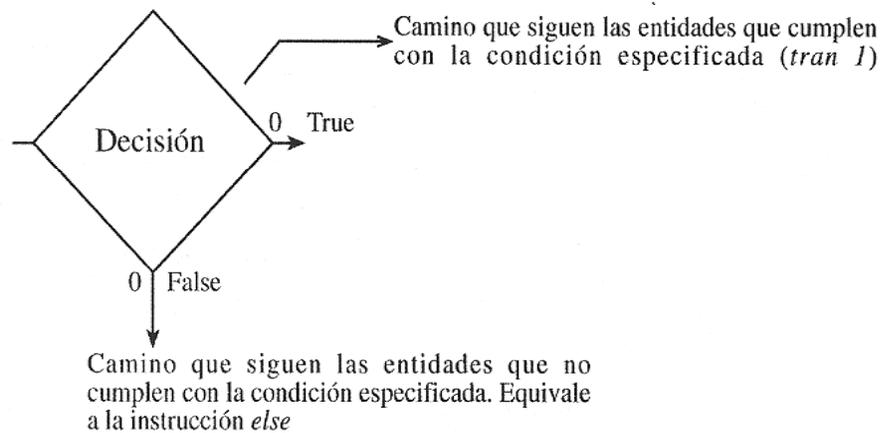
Named:

Entity 1
cliente
Entity 1
tran1
tran2

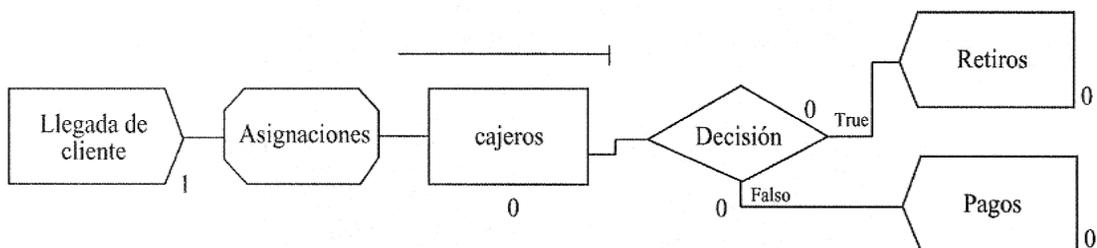
Se cumple la condición; es decir, el atributo *Entity Type* de la entidad que ingresa al módulo es *tran1* (*true*).

Cuando se tienen solamente dos posibles valores para el atributo, se entiende que se toma la segunda condición si no se cumple el primero.

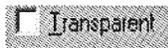
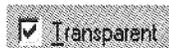
El módulo se ve de la siguiente manera:



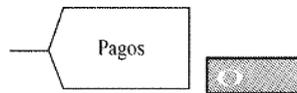
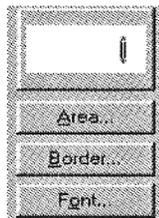
En el caso de que se cuente con más de un valor para un atributo determinado, se escoge la opción *n-way by condition* y se agregan los diferentes *n-1* criterios de escogencia (ejemplo, *tran 1*, *tran 2*, *tran 3*, etc.) teniendo en cuenta que el último no se adiciona debido a que se obtiene por exclusión.



Sugerencia. Se pueden cambiar las propiedades de las animaciones de los contadores (color del área, letras, tamaño, etc.) que se ubican en la parte inferior de algunos módulos, haciendo doble clic en ellas.



Se debe desactivar la casilla de verificación que hace el área transparente para que se habilite la edición del cuadro



3.3 DIVISIÓN DEL FLUJO DE ENTIDADES POR PORCENTAJES

En muchos sistemas productivos, ya sean de servicios o industriales, es posible encontrar situaciones en las cuales el flujo de entidades se puede dividir aleatoriamente, respetando ciertas proporciones que se calculan previamente. Ejemplos clásicos de estas situaciones son los casos que se presentan a continuación:

- El 65% de los clientes en un banco va a realizar retiros o pagos; el 30%, operaciones con CDT'S y el 5% restante va a abrir una nueva cuenta.
- En una inspección, el 5% de los productos presenta algún tipo de no conformidad; el restante pasa la prueba.
- El 40% de los autos que llegan a una estación de servicio requiere combustible diésel; el 20%, gasolina de alto octanaje y el resto, gasolina corriente.

Ejemplo 2.4

■ Descripción

Suponga que se desea modelar la situación del primer caso (ejemplo 2.1). Para ello se muestra la edición del módulo *Decide* dividiendo el flujo de entidades con base en proporciones, donde el 65% de los clientes va a realizar retiros o pagos; el 30%, operaciones con CDT'S y el 5% restante va a abrir una nueva cuenta. Para su mejor comprensión, este módulo se debe insertar en el ejemplo 2.1 bajo las siguientes condiciones:

- Los pagos y retiros son atendidos por un recurso llamado *cajero* (módulo *Process cajeros*, ejemplo 2.1).
- Las operaciones con CDT'S son realizadas por un recurso llamado *asesor financiero*, el cual tiene asociado un tiempo de atención al cliente que sigue una distribución uniforme entre 15 y 30 minutos.
- Las cuentas son abiertas por un recurso llamado *asesor de cuentas* en la oficina de atención al cliente. La apertura de una cuenta de ahorros demora un tiempo que sigue una distribución exponencial con media de 15 minutos.

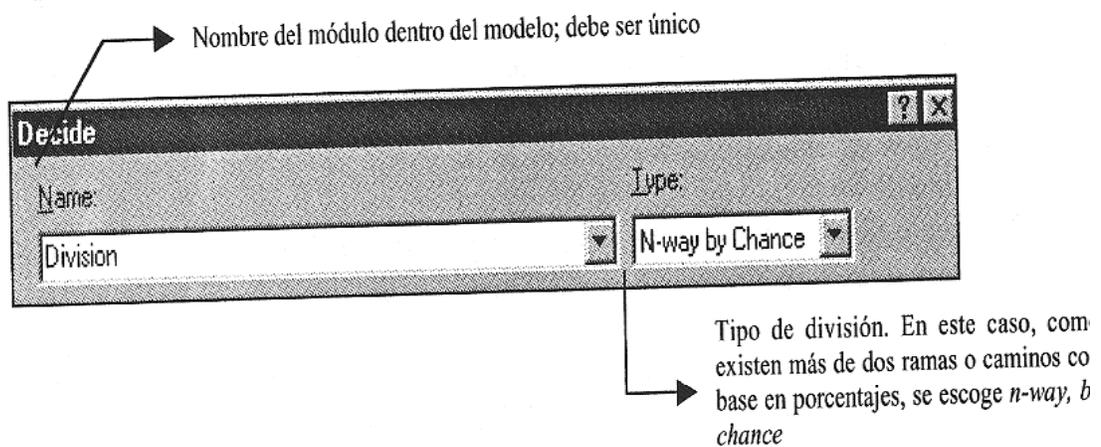
■ Desarrollo

Se emplea un módulo *Decide* para dividir el flujo de entidades por porcentajes.

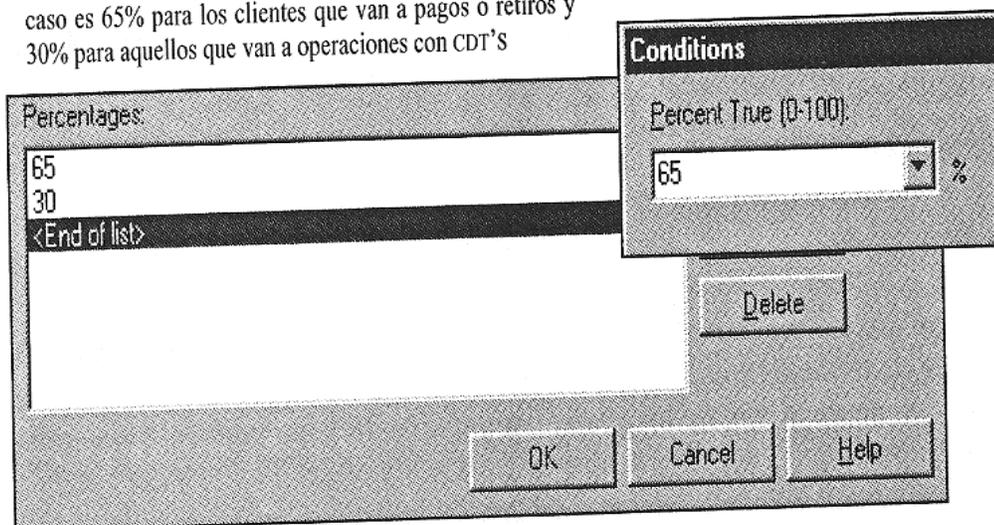


Decide

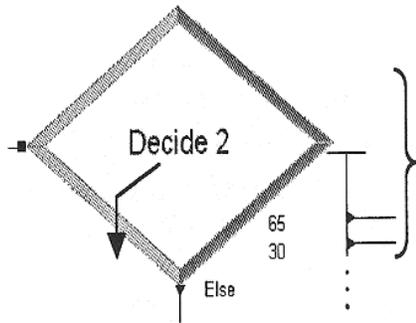
A continuación se muestra la edición del módulo *Decide* para el caso del ejemplo expuesto:



La proporción se expresa como porcentaje real. En este caso es 65% para los clientes que van a pagos o retiros y 30% para aquellos que van a operaciones con CDT'S



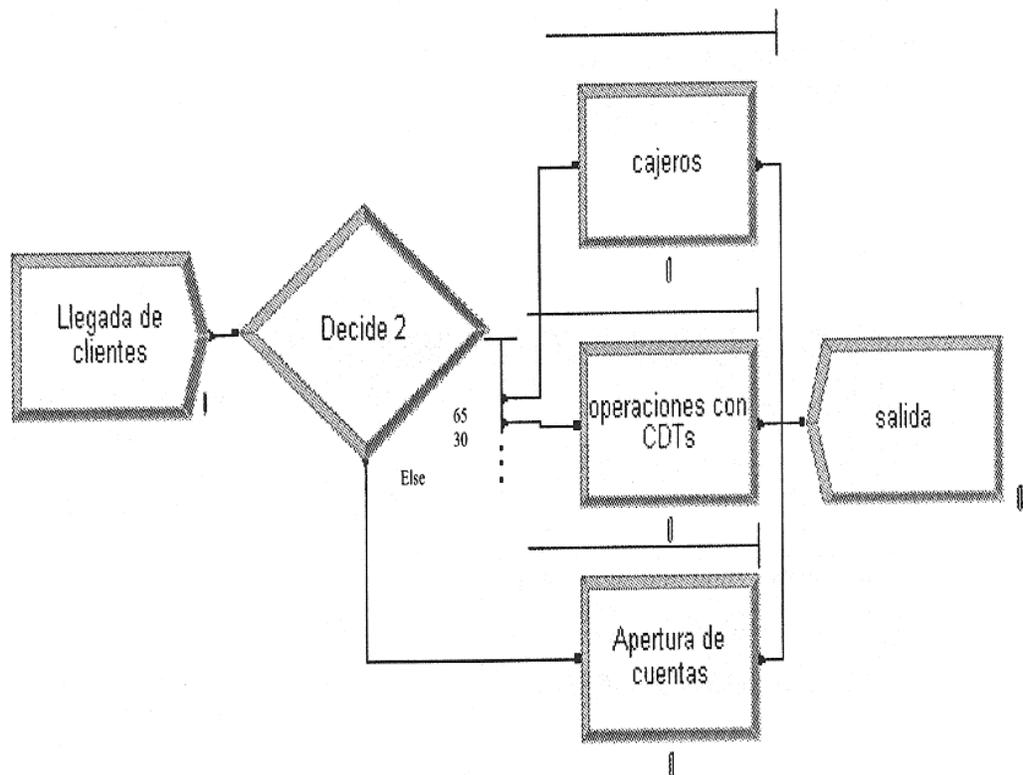
Mediante el botón adicionar se agregan los diferentes porcentajes que componen la decisión



El número de caminos que se sigue, especificados dentro del módulo, debe ser $n-1$, donde n es el total de caminos en los que se divide el flujo. El último camino n se agrega por defecto en la salida *else* del módulo

El último camino n se agrega por defecto en la salida *else* del módulo *Decide* cuando se utiliza *n-way*, ya sea por condición o porcentajes, como sucede en este caso

Para concluir el desarrollo del ejemplo se requiere agregar dos módulos *Process* con los datos especificados anteriormente:



A continuación se muestran los cuadros de diálogo diligenciados para cada módulo *Process* empleado:

Operaciones con CDT'S

- *Apertura de cuentas*

The screenshot shows a 'Process' dialog box with the following fields:

- Name:** Operaciones con CDT's
- Type:** Standard
- Action:** Seize Delay Release
- Priority:** Medium(2)
- Resources:** Resource, Asesor financiero, 1
- Delay Type:** Uniform
- Units:** Minutes
- Allocation:** Value Added
- Minimum:** 15
- Maximum:** 30

Nombre y tipo de módulo *Process*. En este caso se escoge *Standard*, ya que sólo se requiere efectuar un proceso simple; no se necesita modelamiento jerárquico

Acción desarrollada, *seize, delay, release*. Es decir, la entidad ocupa al recurso *asesor financiero*, demora un tiempo determinado y después lo libera

Demora experimentada. En el recurso, debido a la prestación de un servicio, las unidades de tiempo se especifican en minutos

- *Apertura de cuentas corrientes*

The screenshot shows a 'Process' dialog box with the following fields:

- Name:** Apertura de cuentas
- Type:** Standard
- Action:** Seize Delay Release
- Priority:** Medium(2)
- Resources:** Resource, Asesor de cuentas, 1
- Delay Type:** Expression
- Units:** Minutes
- Allocation:** Value Added
- Expression:** EXPO(15)

Nombre y tipo de módulo *Process*. En este caso se escoge *Standard*, ya que sólo se requiere efectuar un proceso simple; no se necesita modelamiento jerárquico

Acción desarrollada, *seize, delay, release*. Es decir, la entidad ocupa al recurso *asesor de cuentas*, demora un tiempo determinado y después lo libera

Demora experimentada. En el recurso, debido a la prestación de un servicio, las unidades de tiempo se especifican en minutos

Para finalizar se conectan todos los módulos *Process* al módulo *Dispose*, que se conoce como salida, en el cual las entidades son retiradas del sistema.

3.4 DIVISIÓN DEL FLUJO DE ENTIDADES CON BASE EN UNA CONDICIÓN DEL SISTEMA

En ocasiones se puede presentar una situación en la cual el flujo de entidades se ve afectado por una condición del sistema. Esta condición se representa por medio de una variable, y su valor numérico simboliza los diferentes estados que puede experimentar el sistema. A continuación se señalan ejemplos de esta situación:

- a. El número de clientes en un sistema de filas de espera en un momento determinado. Esta condición se puede representar con una variable llamada *clientes*, la cual puede adoptar valores de 0, 1, 2, etc.; estos valores definen un estado específico del sistema simulado.
- b. El punto de pedido en un sistema de inventario. La condición del sistema viene determinada por una variable llamada *inventario*; esta variable, al adoptar un valor específico, conduce al sistema a un estado determinado, que en este caso se trataría de un nuevo pedido.
- c. Una señal en un sistema de alimentación de maquinaria. Se puede denominar *ok* a una variable que represente el paso de material de un sitio de espera hasta una máquina específica. Se tiene, entonces, que cuando *ok* adopte el valor 0, las entidades esperan en el sitio correspondiente, y cuando *ok* adopte valor de 1, las entidades se liberan.

Ejemplo 2.5

■ Descripción

Suponga que se desea modelar una situación en la cual los clientes que llegan a un banco y encuentran una fila en la ventanilla de atención con más de 4 personas se retiran sin realizar ninguna transacción.

Para su mejor comprensión, esta situación se debe incluir en el ejemplo 2.1, ya que en éste se tiene una llegada de clientes y un proceso en un recurso llamado *cajero*, el cual cuenta con una fila de espera. Estos aspectos resultan necesarios para desarrollar la situación que se presenta en el ejemplo 2.5.

■ Desarrollo

La entidad que representa al cliente debe tomar una decisión con base en una condición del sistema; para esto se emplea el módulo *Decide*.



Decide

En este módulo es preciso que se pregunte por el número de entidades en la cola del recurso *cajero*. Para esto ARENA[®] cuenta con funciones especiales, las cuales, combinadas con ciertas variables del modelo, dan como resultado una herramienta de gran utili-

dad para el control del flujo de entidades. Una de estas funciones se refiere al número en cola (NQ, *Number in Queue*), cuya sintaxis es como se indica a continuación: NQ (*identificación de la cola*)

Esta función devuelve el número de entidades de la cola especificada en el argumento *identificación de la cola*.

El argumento de la función *identificación de la cola* es el nombre que recibe la espera por la cual se pregunta. Los nombres o identificaciones de las diferentes colas que se encuentran en el modelo se pueden observar en la hoja de trabajo *Queue*, la cual se puede visualizar haciendo clic en el módulo de datos correspondiente en el panel de procesos básicos.



Módulo de datos *Queue*

Queue

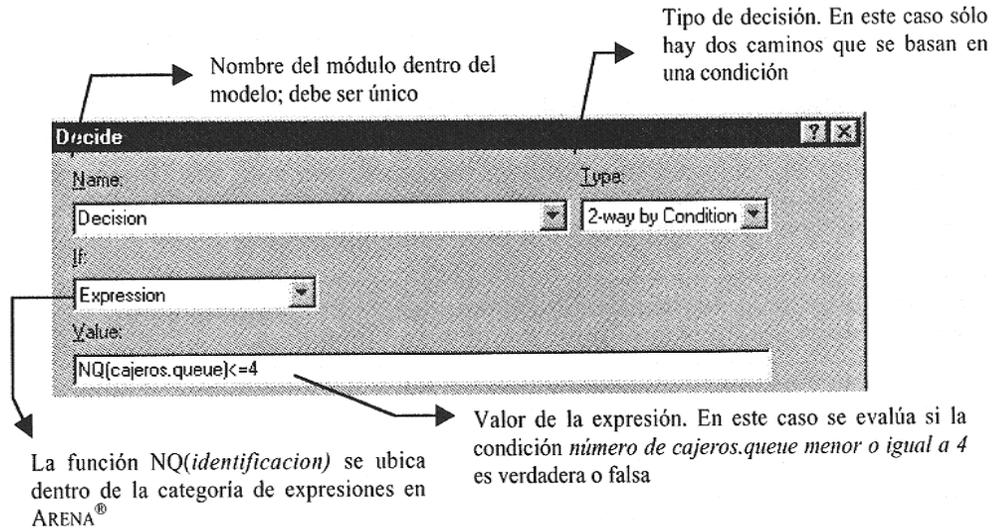
Nombre o identificación de la cola

	Name	Type	Shared
1	cajeros.Queue	First In First Out	<input type="checkbox"/>

Cuando se usa el módulo *Process* bajo una acción *seize*, *delay*, *release*, se crea automáticamente una cola asociada a este módulo, que en este caso recibe el nombre de *Cajeros.queue*. De esta manera, la expresión con la cual se puede conocer el número de entidades en la cola de cajeros es: $NQ(\text{cajeros.queue})^2$.

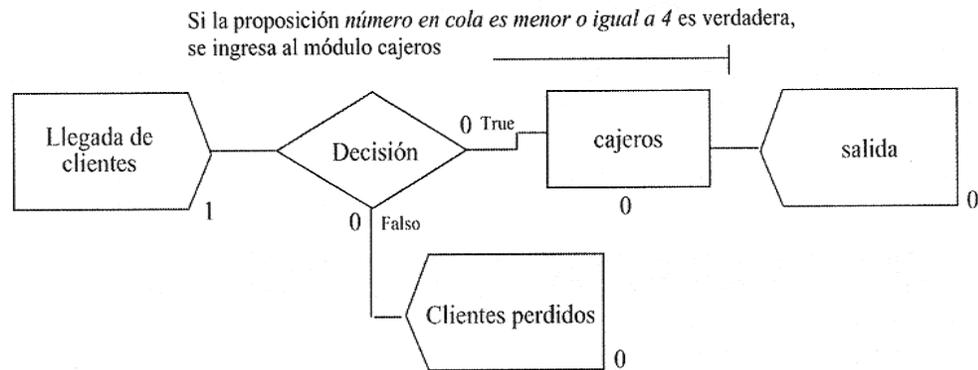
El programa provee otras expresiones útiles. Estas se presentan con más detalle en la guía de variables de ARENA^{® 3}.

El módulo *Decide*, con la expresión formada, se edita de la siguiente manera:



Cuando la condición se da por una expresión, ésta debe incluir el criterio de evaluación, así como se muestra a continuación.

El diagrama de flujo del modelo se verá de la siguiente manera:



Si la proposición *número en cola* es mayor que 4 es falsa, se retira del sistema

Las entidades o clientes que no ingresen al cajero se retiran del sistema mediante un módulo *Dispose*, llamado *clientes perdidos*.

3.5 FORMACIÓN Y DESMEMBRACIÓN DE LOTES O GRUPOS

En procesos industriales en los que se manejan grandes volúmenes de producción es bastante común el uso de lotes o grupos de productos. El tamaño del grupo formado depende de los requerimientos de la operación que se va a realizar. Por ejemplo, puede haber un tamaño de lote para los productos inspeccionados; otro para la transferencia entre máquinas o para la venta o salida.

Los lotes formados, según el tipo, pueden ser desmembrados para procesar individualmente las entidades que lo forman. ARENA® ofrece la posibilidad de formar lotes mediante el módulo *Batch* y de desmembrarlos a través del módulo *Separate*.



Batch



Separate

Se pueden formar dos tipos de lotes: permanentes o temporales; sólo estos últimos se pueden desmembrar para procesar sus entidades componentes.

Importante. Un lote temporal no se puede retirar del modelo sin que antes se desmembre o convierta en permanente; de lo contrario se conducirá a un error en tiempo de ejecución.

3.5.1 Formación de lotes mediante el módulo *Batch*



Batch

El módulo *Batch* permite formar lotes o grupos de entidades. Las entidades que ingresan a este módulo se retienen hasta completar el tamaño indicado para el lote. Cuando se cumple esta condición, una entidad representativa del grupo se libera y continúa a instancias subsiguientes en el modelo.

A continuación se muestra el cuadro de diálogo del módulo *Batch* y se presentan las explicaciones necesarias para su correcta edición:

Nombre del bloque dentro del diagrama de flujo

Tipo de lote que se va a formar; puede ser permanente o temporal

Criterio de asignación de atributos para el lote formado. (Este aspecto se explica posteriormente)

Regla para la formación del lote. Se puede formar para cualquier tipo de entidad (*any entity*) o para entidades con un valor igual de atributo (*by attribute*). En este último caso se debe especificar el atributo en un campo especial que sólo se habilita para esta opción

Tamaño del lote que se va a formar. Según se quiera, puede ser un número o una variable

Criterio de asignación (Save criterion)

El criterio de asignación permite especificar la manera cómo el programa manejará la transferencia de atributos de las entidades individuales hacia el grupo o entidad representativa de éste. El grupo cuenta con los mismos atributos que las entidades que la forman, pero con diferente valor; el cambio de valor obedece a ciertos criterios o reglas.

Los criterios de asignación pueden ser: primero (*first*), último (*last*), suma (*sum*) y producto (*product*). En la siguiente tabla se explica la función de cada uno:

Criterio	Descripción
<i>Primero (first)</i>	Asigna a los atributos de la entidad representativa los valores de los atributos de la primera entidad que llegó al módulo <i>Batch</i> .
<i>Último (last)</i>	La entidad representativa toma los valores de los atributos de la última entidad que llegó al módulo <i>Batch</i> , es decir, la que completa el tamaño del lote.
<i>Suma (sum)</i>	El valor de los atributos de la entidad representativa se forma a partir de la suma de los valores de los atributos de las entidades que conforman el lote. Por ejemplo, si se tiene un atributo llamado <i>tipo</i> para las entidades individuales, la entidad representativa contendrá un atributo llamado <i>tipo</i> pero, en este caso, su valor será equivalente a la suma de los atributos del mismo nombre de las entidades individuales.
<i>Producto (product)</i>	Este criterio funciona de manera análoga al de la suma; en este caso, el valor del atributo para la entidad representativa proviene de la multiplicación de los valores de los atributos del mismo nombre de las entidades individuales.

Importante. Cuando se forman lotes permanentes, el atributo *Entity.CreateTime* se transfiere con el criterio *first*, sin importar que se estén usando otras reglas, como *sum*, *product* o *last*.

3.5.2 Desmembración de lotes mediante el módulo *Separate*

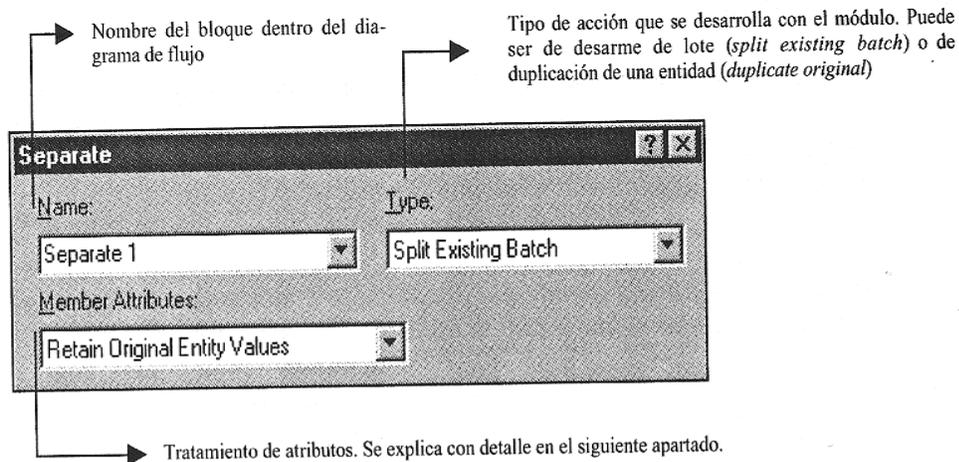


Separate

El módulo *Separate* se usa para desmembrar un lote temporal en las entidades que lo componen; adicionalmente, tiene la función de duplicar entidades (este aspecto se estudia posteriormente).

El funcionamiento de este módulo es bastante simple. La entidad que representa al lote entra al módulo *Separate*, e inmediatamente después se desmembra para convertirla en las múltiples entidades que la forman.

A continuación se muestra el cuadro de diálogo del módulo *Separate* y se presentan las explicaciones necesarias para su correcta edición en el desarme de lotes:



Tratamiento de atributos (Member attributes)

Cuando se desmembra un lote, al igual que cuando se forma, se debe establecer una regla o criterio para la transferencia de los valores de los atributos de la entidad representativa hacia las entidades componentes. Estas reglas se explican a continuación:

Criterio	Función	Descripción
<i>Retain original entity values</i>	Retener valores originales	Con esta opción, cuando se desmembra el lote, las entidades conservan los valores que originalmente tenían sus atributos correspondientes.
<i>Take all representative values</i>	Asignar todos los valores representativos	Esta opción asigna los valores de los atributos de la entidad representativa a todas las entidades componentes.
<i>Take specific representative values</i>	Asignar sólo valores representativos específicos	Esta opción permite asignar sólo los valores de los atributos de la entidad representativa o lote a ciertos atributos de las entidades individuales. Estos atributos se especifican en un cuadro de lista que se habilita cuando se escoge esta opción. Los atributos no seleccionados conservan sus valores originales después del desarme del lote.

Ejemplo 2.6

■ Descripción

Suponga un proceso manufacturero mediante el cual se fabrica un determinado tipo de pieza para la industria automotriz; las partes sin procesar llegan con una distribución exponencial con media de 5 minutos.

Las piezas son procesadas en un torno en un tiempo uniforme entre 3 y 6 minutos, y al terminar son almacenadas hasta formar un grupo de 20 unidades. Posteriormente pasan a un horno; en éste se efectúa un tratamiento térmico al lote en un tiempo constante de 20 minutos. Una vez fuera del horno, las piezas son pulidas una a una en un tiempo constante de 2 minutos por elemento y empacadas en cajas de 10 unidades.

Para explicar el funcionamiento de los módulos *Batch* y *Separate*, se crean dos atributos de prueba llamados *Atributo A* y *Atributo B*, con valores de 1 y 2, respectivamente.

Nombre	Valor
Atributo A	1
Atributo B	2

Cuando se forma el primer lote de transferencia, los valores de los atributos de la entidad representativa son la suma de los atributos de las entidades componentes. Cuando se desarma, sólo el *Atributo A* conserva los atributos del lote o entidad representativa.

■ Desarrollo

El primer paso es generar las llegadas adicionando un módulo *Create* llamado *llegada de partes*. A continuación se muestran los campos que se han editado de este módulo tabulados en su respectiva hoja de trabajo:



Create

	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Llegada de partes	partes	Random	5	Minutes	1	Infinite	0.0

Luego se procede a asignar los atributos antes mencionados por medio de un módulo *Assign*:



Assign

Número de filas que componen el módulo. Cada fila es equivalente a una asignación

Si se hace clic en el botón que indica el número de filas (*rows*), se pueden ver las asignaciones en una hoja de trabajo aparte, lo cual da la oportunidad de editar e incluso de realizar una nueva

	Name	Assignments
1	Asignacion de atributos	rows

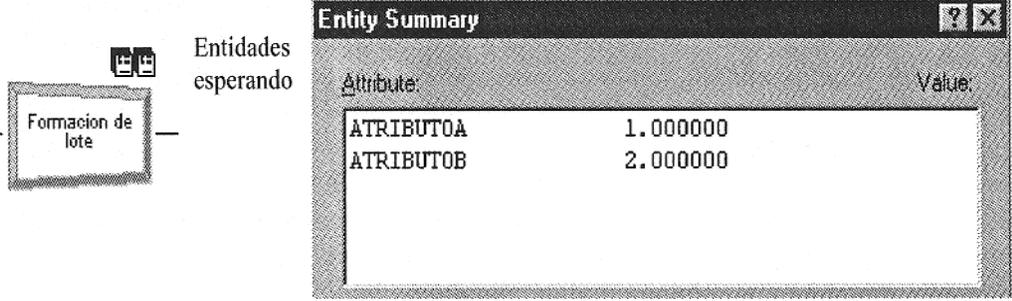
Hoja de trabajo que contiene las asignaciones hechas en el módulo. Si se sigue la instrucción que está al final de las columnas, se puede hacer doble clic para realizar una nueva asignación

Assignments			
	Type	Attribute Name	New Value
1	Attribute	atributoA	1
2	Attribute	atributoB	2

Double-click here to add a new row.

Ahora todas las entidades cuentan con dos atributos: el *Atributo A*, con valor 1, y el *Atributo B*, con valor 2.

Sugerencia. Es posible examinar las entidades que se encuentran en las colas o en proceso haciendo una pausa en la ejecución del modelo mediante la tecla *Esc* o el botón , y doble clic en la entidad elegida para acceder al cuadro de diálogo *Entity Summary*.



Atributo:	Value:
ATRIBUTOA	1.000000
ATRIBUTOB	2.000000

La operación de turno se desarrolla mediante un módulo *Process*: se diligencia su cuadro de diálogo o su hoja de trabajo con los datos que se presentan en el enunciado del ejemplo.

En la salida del proceso de turno se coloca un módulo *Batch*, con el propósito de formar el lote de transferencia que se pide en el ejemplo.

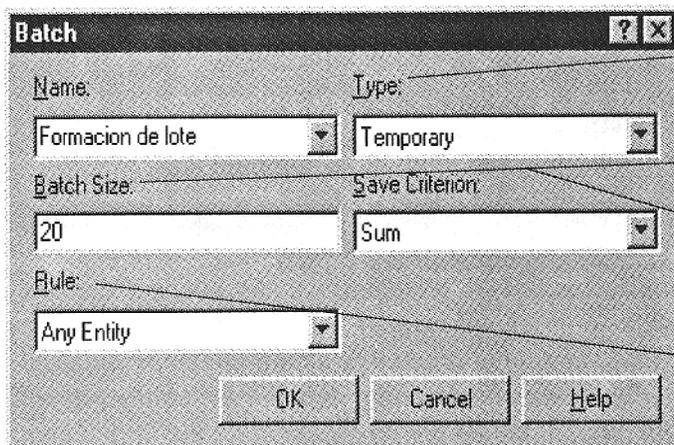


Process



Batch

	Name	Type	Action	Priority	Resource	Delay Type	Units	Allocation	Minimum	Value	Maximum
1	Torno	Standard	Seize Delay Release	Medium	1 rows	Uniform	Minutes	Value Added:3			6



Cuadro de diálogo del módulo *Batch*. En éste se especifican el tipo de lote, que en este caso es temporal

Tamaño del lote que se va a formar, 20

Criterio de asignación para los atributos representativos. En este caso viene determinado por la suma de los atributos componentes

Regla de formación del lote. Viene determinada para cualquier tipo de entidad

Los atributos que resultan para la entidad representativa son los siguientes:

Atributo	Valor
Atributo A	20
Atributo B	40

Posteriormente el lote pasa al horno, el cual se representa con un módulo *Process* sólo con acción de demora *delay*.

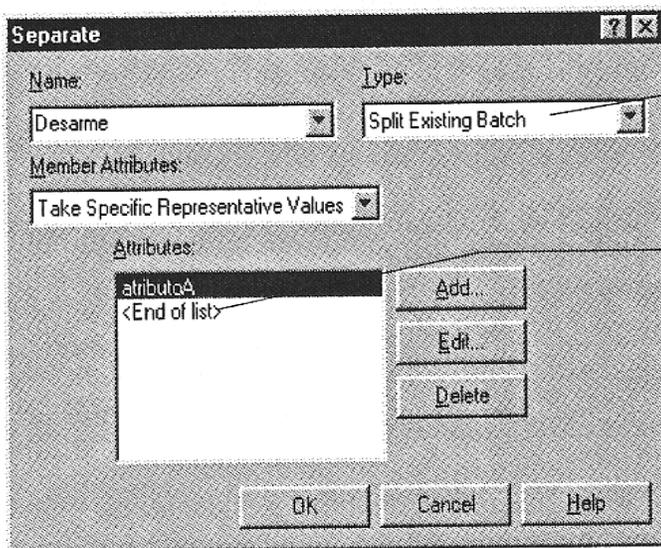


Process

Luego de pasar por el horno, el lote es desarmado con un módulo *Separate*:

	Name	Type	Action	Priority	Resource	Delay Type	Units	Allocation	Minimum	Value	Maximum
1	Torno	Standard	Seize Delay Release	Medium	1 rows	Uniform	Minutes	Value Added	3		6
2	Horno	Standard	Delay	Medium	1 rows	Constant	Minutes	Value Added	20		

Separate



Cuadro de diálogo del módulo *Separate*.

El tipo de acción que se debe seguir es desarmar un lote existente (*split existing batch*)

En lo que respecta al tratamiento de atributos, sólo se conservan valores específicos del lote. De acuerdo con lo que se indica en el enunciado, sólo se conserva el *Atributo A*; el resto conserva su valor original

Cuando se desmembra el lote y se especifica que sólo se conservan los valores representativos del *Atributo A*, se tienen los siguientes nuevos valores:

Atributo	Valor
Atributo A	10
Atributo B	2

El *Atributo B* conserva el valor que tenía originalmente, mientras que el *Atributo A* permanece con el valor que se le asignó para el lote.

Después que las piezas están desmembradas se procesan en la operación de pulido. La hoja de trabajo respectiva se muestra a continuación:



Process

	Name	Type	Action	Priority	Resource	Delay Type	Units	Allocation	Minimum	Value	Maximum
1	Torno	Standard	Seize Delay Release	Medium	1 rows	Uniform	Minutes	Value Added	3		6
2	Horno	Standard	Delay		1 rows	Constant	Minutes	Value Added		20	
3	Pulido	Standard	Seize Delay Release	Medium	1 rows	Constant	Minutes	Value Added	5	2	5

Finalmente, para empaclar las piezas en cajas de 10 unidades, se emplea un módulo *Batch* de tipo permanente.



Batch

	Name	Type	Batch Size	Save Criterion	Rule
1	Formacion de lote	Temporary	20	Sum	Any Entity
2	Empaque final	Permanent	10	Last	Any Entity

Una vez empaçadas, las entidades están listas para dejar el sistema mediante un módulo *Dispose*.



Dispose

3.6 CREACIÓN DE VARIAS ENTIDADES A PARTIR DE UNA ENTIDAD

En muchos casos, tanto en el medio industrial como en el de servicios, es posible encontrar situaciones en las cuales se requiere duplicar una entidad, la cual puede ser una pieza o una orden de pedido. Como ilustración de la situación en estudio considere los siguientes ejemplos:

- En un proceso de troquelado entra una lámina de dimensiones específicas a la máquina, y salen seis arandelas como producto terminado o subproducto.
- En un laboratorio clínico, a un paciente se le extrae una muestra de sangre, la cual se identifica y examina. El cliente pasa a una sala de espera para la entrega de resultados. En este caso, la muestra es un duplicado del paciente, la cual espera su turno para ser examinada por los bacteriólogos.

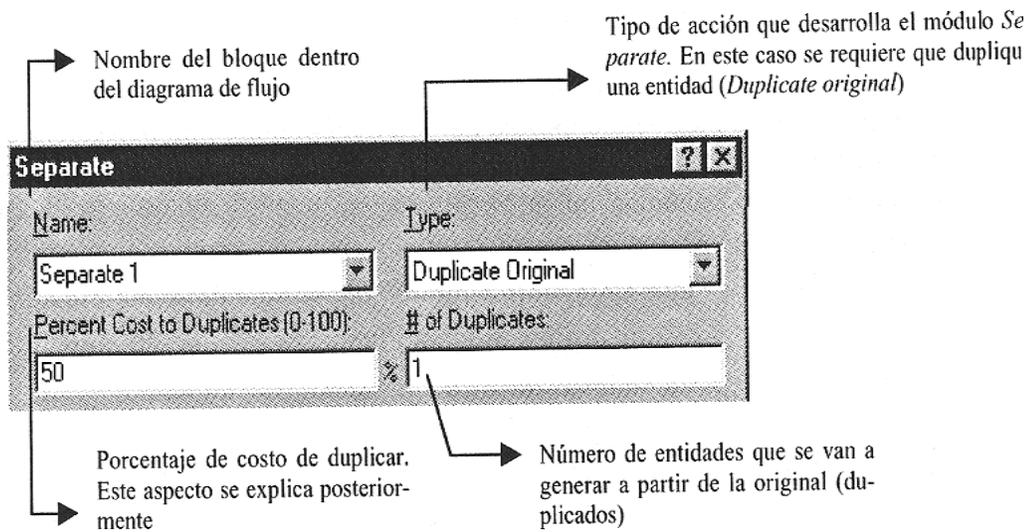
- c. En una compañía de distribución, una orden de pedido que puede ser procesada por dos departamentos al mismo tiempo (procesamiento en paralelo) debe ser duplicada, procesada y consolidada al final.

La función en estudio se ejecuta en ARENA® mediante el módulo *Separate*. Este módulo se usa también para la separación o desarme de lotes temporales o grupos.

A continuación se muestra el cuadro de diálogo del módulo *Separate* para la función de duplicar entidades:



Separate



El módulo *Duplicate* funciona de manera similar a una fotocopidora: entra la entidad original y sale el número de copias o duplicados que se especifiquen.

Importante. Note que cuando se duplican entidades se tiene el número de duplicados +1 (entidad original) entidades en el sistema.

Porcentaje del costo de duplicar

Este campo se llena con un número que va entre 0 y 100, el cual indica el porcentaje de los costos y el tiempo de la entidad original que se distribuye equitativamente entre el número de duplicados y la entidad original; mientras tanto, la entidad original conserva el porcentaje restante.

Considere como ejemplo un caso en el cual el porcentaje del costo de duplicado es de 90% y el número de entidades que se va a duplicar es 3. La situación se maneja como sigue:

- El 90% del costo se reparte uniformemente entre los duplicados, es decir, 30% para cada uno.
- El 10% restante se asigna a la entidad original.

Esta asignación se realiza sólo para los atributos de costo y tiempo, los cuales se muestran a continuación:

Atributos de costos	Atributos de tiempo
Entity.VACost	Entity.VATime
Entity.NVACost	Entity.NVATime
Entity.WaitCost	Entity.WaitTime
Entity.TransferCost	Entity.TransferTime
Entity.OtherCost	Entity.OtherTime

Los atributos de propósito especial y los que define el usuario no se distribuyen según el criterio anteriormente expuesto, sino que se duplican como una copia del atributo original, tanto en identificación como en valor. Los atributos de propósito especial son los que indican la secuencia, la estación en la cual se encuentra la entidad, etc. A continuación se presenta una lista de éstos:

Atributos de propósito especial
Entity.SerialNumber
Entity.Type
Entity.Jobstep
Entity.Picture
Entity.Station
Entity.Sequence
Entity.HoldCostRate

Importante. El atributo de propósito especial *Entity.CreateTime*, que almacena el tiempo en el cual una entidad es creada en el modelo y que es empleado para determinar el tiempo en el sistema, no se transfiere a los duplicados. Este atributo toma el valor del tiempo en que las entidades se crean; en este caso se duplican (TNOW).

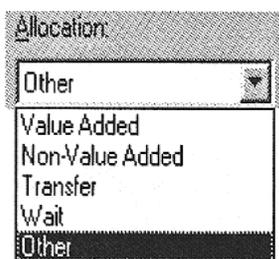
TRATAMIENTO DE COSTOS

Cuando se desarrolla una actividad productiva se incurre en ciertos costos, los cuales varían según la naturaleza de esa actividad. En general, un costo se puede originar en actividades que dan o no valor agregado al producto que se fabrica o al servicio que se presta; esto puede ser, de acuerdo con el sistema estudiado, un costo de espera o transferencia, o una categoría especial. En ARENA[®] es posible clasificar los costos, según las operaciones que se realicen, mediante una herramienta que viene incorporada en esta versión del programa.

Las categorías de costo en ARENA[®] pueden ser:

- Costos de valor agregado (*Value added cost, VAcost*)
- Costos que no agregan valor (*Non value added cost, NVAcost*)
- Costos de transferencia (*Transfer cost*)
- Costos de espera (*Wait cost*)
- Otros costos (*Other cost*)

El usuario debe establecer previamente la clasificación de las actividades que generan estos costos e ingresarla posteriormente al modelo en los campos especialmente diseñados para este propósito. Estos campos reciben el nombre de *Allocation*, e indican la ubicación dentro de las categorías antes mencionadas; a esta ubicación pertenece el costo de la actividad realizada.



El campo *Allocation* aparece en todos los módulos que puedan afectar el tiempo de una entidad en el sistema. Su función es asignar el tiempo que la entidad pasa en este módulo a alguna de las categorías de costo

Es importante aclarar que los costos se calculan con base en el tiempo que demora la entidad en actividades que se encuentran asociadas con alguna de las categorías de costo.

1. COSTO INICIAL

ARENA[®] ofrece la posibilidad de asignar una determinada cantidad a una categoría de costo antes de comenzar la corrida. Para cada tipo de entidad este valor inicial se define en la hoja de trabajo *Entity*, la cual se muestra a continuación:

Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran Cost	Initial Other Cost
Entity 1	Picture Rep	0.0	0.0	0.0	0.0	0.0	0.0

Double-click here to add a new row.

En esta hoja de trabajo se pueden inicializar valores para cada categoría de costo de acuerdo con el tipo de entidad (*Entity Type*).

2. COSTO DE PERMANENCIA

A menudo se requiere calcular el costo en el que se incurre por mantener una entidad en el proceso o en una parte específica de éste. Esto se conoce como costo de permanencia de una entidad en el sistema por unidad de tiempo (ya sea en proceso o en espera).

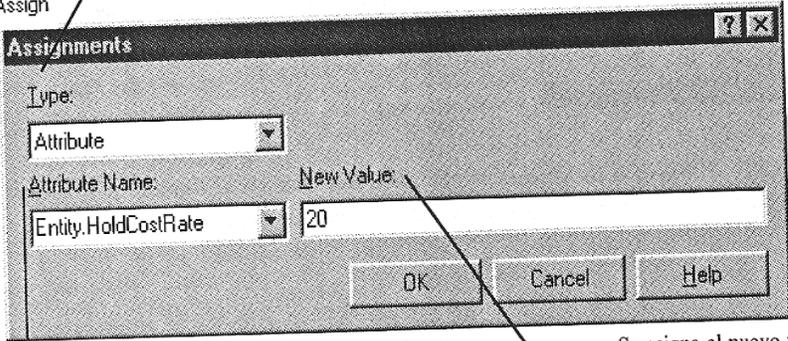
El costo por unidad de tiempo en el sistema o costo de permanencia se almacena en un atributo llamado *HoldCostRate*, que se define inicialmente en la hoja de trabajo de la entidad y se expresa en unidades monetarias por hora.

El costo de permanencia lo asigna el usuario; es decir, su valor se puede modificar mediante módulos lógicos y de acuerdo con las necesidades del analista. Otro aspecto que es importante destacar es que este costo es un atributo, por lo tanto puede tomar valores distintos para cada entidad, ser una distribución, formar grupos (*sets*), en general, puede desarrollar todas aquellas aplicaciones propias de los atributos.

El valor del costo de permanencia en tiempo de corrida se cambia mediante el módulo *Assign*:

Assign

En general, cuando se cambia el valor del costo de permanencia, se escoge el tipo de asignación atributo *Attribute*



Del menú desplegable se selecciona el atributo que almacena el costo de permanencia por hora en el sistema para cada entidad *Entity.HoldCostRate*

Se asigna el nuevo valor. Mientras éste no modifique, el tiempo (en horas) que demora la entidad en el sistema tendrá un costo de U.M. sin importar si se procesa o no

3. COSTO DE PROCESAMIENTO

Se incurre en un costo de procesamiento cuando una entidad hace uso de un recurso. Este costo se asigna a la categoría que el usuario decida de acuerdo con sus necesidades y con la organización del modelo.

En un recurso es posible tener los siguientes costos:

- *Costo de recurso ocioso.* Costo por hora en el cual el recurso no experimenta un estado de falla ni de ocupación; simplemente ninguna entidad lo está ocupando.
- *Costo de recurso ocupado.* Costo por hora de uso de un recurso determinado.
- *Costo por uso.* Costo por unidad procesada en el recurso. Se incurre en este costo cada vez que una entidad toma control de una unidad de recurso.

Estos costos se definen en la hoja de trabajo del recurso *Resource*, así como se muestra a continuación:

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	Resource 1	Fixed Capacity	1	0	0	0		0 rows

Double-click here to add a new row.

De estas categorías, sólo los costos por uso (*per use*) y por hora de ocupación (*Busy/hour*) se cargan a la entidad. El costo por hora ociosa (*Idle/hour*) se muestra en los reportes de costos del recurso, pero no hace parte del costo de procesamiento de la entidad.

Adicionalmente, se carga el de permanencia. Si éste es mayor que 0, el tiempo que demore la entidad en el recurso tendrá un costo que es determinado por el valor del atributo *Entity.HoldCostRate*, que se carga a la categoría de costo a la cual pertenece la actividad que se desarrolla en el recurso.

Importante. El costo generado por el tiempo que demora una entidad en espera de la atención de un recurso es asignado, por defecto, a la categoría WAIT y evaluado con el atributo *Entity.Hold.CostRate* (si es mayor que 0).

Ejemplo 3.1

■ Descripción

Llega una pieza a un proceso que consta de tres operaciones:

Operación	Tiempo (minutos)
Fresado	Uniforme (4,8)
Taladrado	Normal (5,89,1.5)
Pulido	Triangular (3,5,7)

Las partes llegan al proceso con un tiempo exponencial con media de 5 minutos. Las entidades llegan con un costo acumulado de valor agregado de 3 U.M.

En la máquina de fresado se incurren en los siguientes costos:

- Costo por hora de uso de 5 U.M.
- Costo por pieza procesada de 2 U.M.
- Costo de tenencia de la pieza en la máquina de fresa (espera y proceso) de 1.5 U.M.

En la máquina de taladro se incurren en los siguientes costos:

- Costo de máquina ociosa de 3 U.M.
- Costo por pieza elaborada de 2 U.M.
- Costo por uso de la máquina de 1.3 U.M.

El costo de tenencia de la pieza es 0

En la máquina de pulido se incurren en los siguientes costos:

- Costo de máquina ociosa de 4 U.M.
- Costo por pieza elaborada de 1 U.M.
- Costo por uso de la máquina de 3 U.M.

El costo de tenencia de la pieza es 0

Adicionalmente, las piezas experimentan una demora uniforme entre 5 y 20 minutos. Esta demora no agrega valor al producto, y sus circunstancias hacen que se incurra en un costo de tenencia de 11 U.M. Finalmente, las entidades se retiran del sistema.

Se debe determinar el costo promedio por pieza, el costo total, así como los costos de valor agregado, de no valor agregado y de esperas.

■ Desarrollo

a. Entrada de entidades

El punto de partida es la creación de las entidades según el tiempo entre llegadas que se indica en el enunciado; para esto se emplea el módulo *Create*.



Create

Nombre del módulo y nombre del atributo *Entity Type* para identificar las entidades que ingresan por el módulo *Create*

Tipo de llegada. Cuando es exponencial se escoge *Random*, y en el campo *Value* se coloca el valor de la media. Además, se definen las unidades

Valores por defecto para el tamaño del lote de llegada (*Entities per arrival*), número máximo de llegadas (*Max arrivals*) y tiempo de creación de la primera entidad (*First creation*)

b. Operación de fresado

Luego de crear el punto de entrada se agrega la primera operación, que corresponde a fresado. Para esto se emplea un módulo *Process*:

Se emplea un módulo *Process* tipo estándar, que recibe el nombre de fresado

Se emplea un recurso de nombre *fresa* (diferente al nombre del módulo)

Se indica la categoría a la cual se desea que se carguen los costos en los que se incurra durante esta actividad

Se ingresan los datos de la demora en minutos, según las indicaciones del enunciado.

En este punto se deben ingresar los costos de procesamiento. El costo de permanencia en el recurso se establece en la hoja de trabajo de la entidad, ya que ésta es la primera operación:

Costo de permanencia por hora

Costo inicial en la categoría de valor agregado

Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran	
1	Entidad1	Picture.Report	1.5	3	0.0	0.0	0.0

Double-click here to add a new row.

Según las condiciones del ejercicio, en la hoja de trabajo también se ingresa el valor inicial del costo en la categoría de valor agregado.

Los costos relacionados con el recurso, como costo por uso y por tiempo de ocupación, se ingresan en la hoja de trabajo del recurso, así como se muestra a continuación:

Resource

Costo por hora de ocupación

Costo por uso

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	Fresa	Fixed Capacity	1	5	0.0	2		0 rows

Double-click here to add a new row.

c. Operación de taladrado

Antes de agregar la operación de taladrado se debe tener en cuenta que el costo de tenencia de las partes es 0; es decir, no se incurre en ningún costo por este motivo. Esta condición implica que se debe asignar un valor de 0 al atributo *Entity.HoldCostRate*, el cual conserva el valor inicialmente asignado para la operación de fresado. Para eso se emplea un módulo *Assign*:

Assign

Así como se indicó anteriormente, el valor del atributo que almacena el costo por hora de permanencia de entidades en el sistema (*Entity.HoldCostRate*) se cambia; en este caso se cambia por 0. Con esto se pretende señalar que no se va a incurrir más en ese costo

Assignments [?] [X]

Type: Attribute

Attribute Name: Entity.HoldCostRate

New Value: 0

OK Cancel Help

Una vez se descarga este costo se procede a agregar la operación de taladrado mediante la inclusión de un módulo *Process*, editándolo según las condiciones del enunciado. A continuación se muestra esta operación:

Se emplea un módulo *Process* tipo estándar, que recibe el nombre de *taladrado*

Se emplea un recurso que tiene el nombre de *taladro* (diferente al nombre del módulo)

Se indica la categoría a la cual se desea que se carguen los costos en los que se incurra durante esta actividad

Se ingresan los datos de la demora en minutos, según las indicaciones del enunciado

Para agregar los costos por procesamiento se emplea la hoja de trabajo de los recursos, a la cual se accede mediante el módulo de datos *Resource*.



Resource

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	Fresa	Fixed Capacity	1	5	0.0	2		0 rows
2	Taladro	Fixed Capacity	1	1.3	3	2		0 rows

Se introducen los datos de costo por hora de uso, por hora ociosa y por pieza procesada

d. Operación de pulido

Con el propósito de seguir la secuencia, se adiciona la operación de pulido. Cuando no se incurre en costos de tenencia, sólo es necesario adicionar un módulo *Process* para desarrollar la operación, ya que el atributo *Entity.HoldCostRate* conserva el valor de 0 que se le asignó en instancias anteriores.



Process

Se emplea un módulo *Process* tipo estándar, que recibe el nombre de *pulido*

Se emplea un recurso que tiene el nombre de *pulidora* (diferente al nombre del módulo)

Se indica la categoría a la cual se desea que se carguen los costos en los que se incurra durante esta actividad

Se ingresan los datos de la demora en minutos, según lo indica el enunciado

Los costos respectivos de la operación de pulido se ingresan al modelo en la hoja de trabajo para los recursos, a la cual se accede mediante el módulo de datos *Resource*. A continuación se muestra este paso:



Resource

Se introducen los datos de costo por hora de uso, por hora ociosa y por pieza procesada

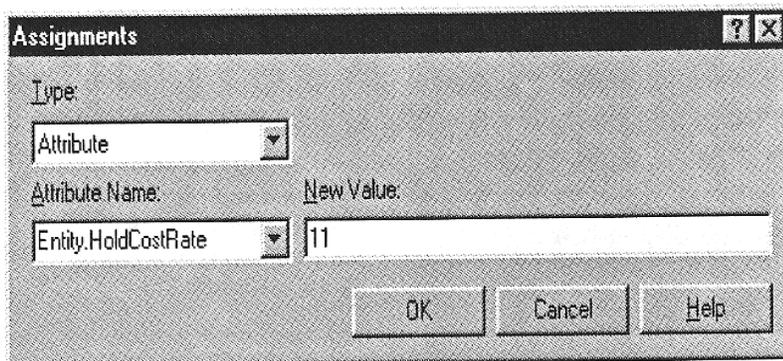
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	Fresa	Fixed Capacity	1	5	0.0	2		0 rows
2	Taladro	Fixed Capacity	1	1.3	3	2		0 rows
3	pulidora	Fixed Capacity	1	3	4	1		0 rows

e. Demora

Como se indica en el enunciado, las entidades deben experimentar una demora que no agrega valor pero sí carga costos de tenencia. Por ello se debe modificar el atributo *Entity.HoldCostRate* con el nuevo valor; para esto se emplea un módulo *Assign*, así como se presenta a continuación:



Assign



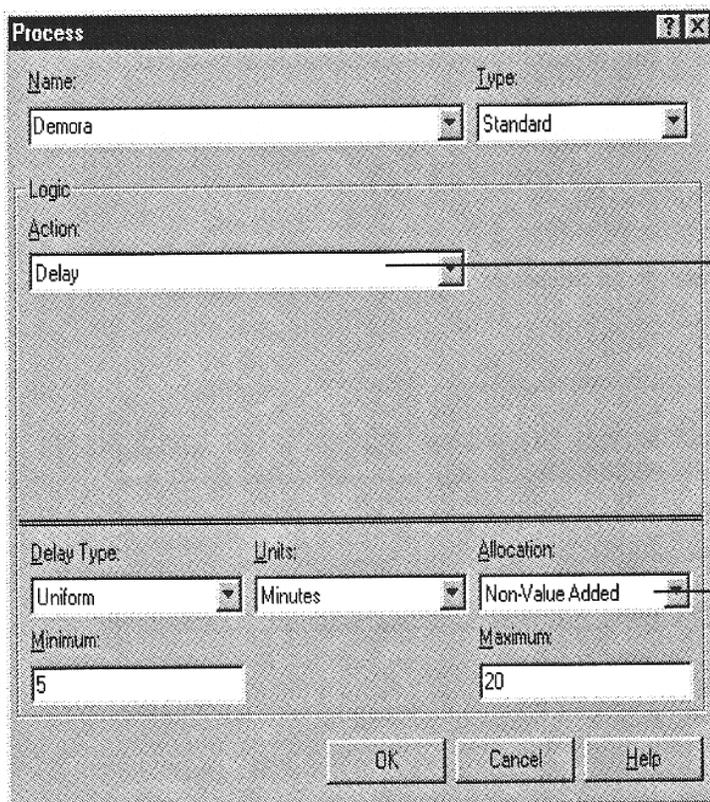
The 'Assignments' dialog box has a title bar with a question mark and a close button. It contains a 'Type' dropdown menu set to 'Attribute'. Below it are two fields: 'Attribute Name' with a dropdown menu set to 'Entity.HoldCostRate' and 'New Value' with a text input field containing '11'. At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

Al atributo *Entity.HoldCostRate* se le asigna el valor del costo por hora de tenencia de entidades para la demora

Una vez se ha modificado el valor del costo de tenencia, y según las condiciones del enunciado, se agrega un módulo *Process* con la acción configurada como *delay* y clasificada como de no valor agregado.



Process



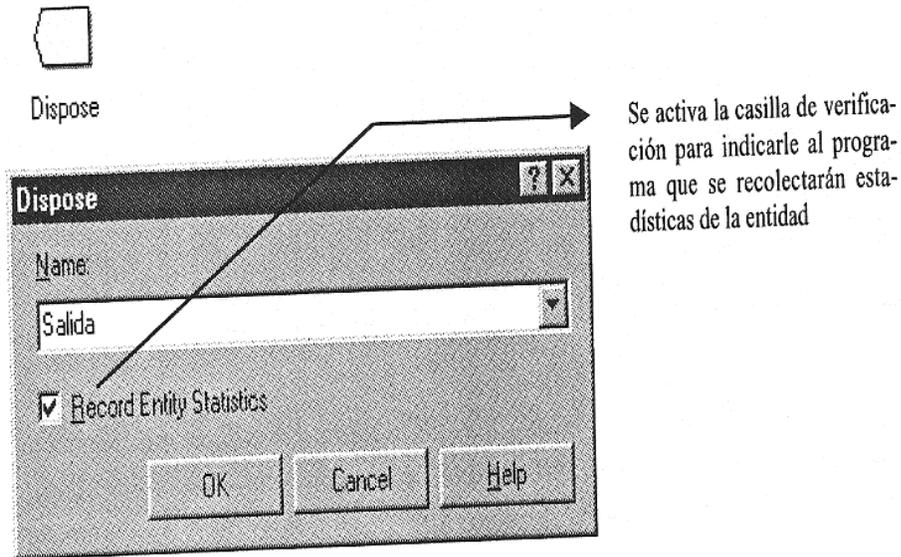
The 'Process' dialog box has a title bar with a question mark and a close button. It contains a 'Name' dropdown menu set to 'Demora' and a 'Type' dropdown menu set to 'Standard'. Below is a 'Logic' section with an 'Action' dropdown menu set to 'Delay'. At the bottom are three fields: 'Delay Type' set to 'Uniform', 'Units' set to 'Minutes', and 'Allocation' set to 'Non-Value Added'. Below these are 'Minimum' and 'Maximum' text input fields containing '5' and '20' respectively. At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

Se configura la acción del módulo *Process* sólo como demora (*delay*). No involucra la utilización de un recurso

Se establece la duración de la acción y se clasifica su costo según lo indica el enunciado

f. Salida de entidades

Una vez se cumplió la demora, las entidades se deben retirar del sistema, y sus estadísticas se deben recolectar. Esto se desarrolla mediante el módulo *Dispose*:

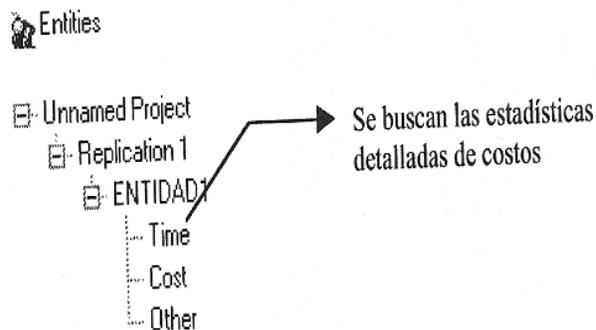


g. Corrida del modelo y análisis de resultados

En el caso de que se corra una réplica del modelo durante 2.400 minutos, se puede hacer un análisis de los costos de producción por pieza y por utilización de recursos, así como detallar la categoría de cada uno, por ejemplo, valor agregado, espera, etc.

En los reportes de estadísticas de las entidades *Entities* se pueden apreciar los costos por tipo de entidad en sus diferentes categorías. Para su fácil manejo se debe emplear el explorador de reportes.

El primer reporte de interés es el de las entidades:



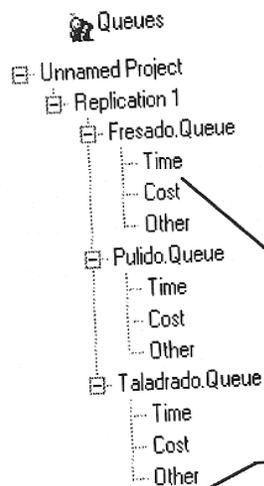
En la ventana se mostrará el reporte indicado:

Cost	Average	HalfWidth	Minimum	Maximum
NVA Cost	134.90	5.83315	55.6798	219.24
Other Cost	0.00	0.000000000	0.00	0.00
Total Cost	148.56	5.96452	65.6204	232.84
Transfer Cost	0.00	0.000000000	0.00	0.00
VA Cost	9.0271	0.017448311	8.7273	9.3449
Wait Cost	4.6313	(Correlated)	0.00	9.7858

En el reporte aparecen los costos por categoría en los que incurrió la entidad

Existen diferentes clases de reportes de acuerdo con el grado de detalles que se quiera averiguar; los menos detallados se encuentran en los primeros niveles cuando se navega por los reportes.

Otro reporte que puede ser de particular interés es el de las colas, *Queues*:



En el navegador de reportes se escoge el nivel de detalles y el tipo de información que se desea observar

Para el caso del ejemplo, suponga que se quieren conocer los costos de espera en la cola de fresado. (Es la única operación con recurso que tiene un valor de *Entity.HoldCostRate* mayor que 0)

En la ventana aparece el reporte correspondiente

Replication 1	Start Time:	0.00	Stop Time:	2,400.00	Time Units:	Minutos
----------------------	-------------	------	------------	----------	-------------	---------

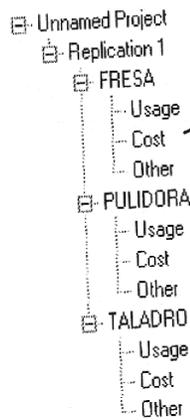
Fresado.Queue					
Time	Average	HalfWidth	Minimum	Maximum	
Waiting Time	188.78	(Correlated)	0	392.34	
Cost	Average	HalfWidth	Minimum	Maximum	
Waiting Cost	4.7194	(Correlated)	0	9.8086	
Other	Average	HalfWidth	Minimum	Maximum	
Number Waiting	38.5941	(Correlated)	0	87.0000	

Se observan las estadísticas de costos de espera de la cola de fresado, así como también las de tiempo y número en cola

Finalmente, existen costos asociados a la utilización de los recursos que, según sea el caso, pueden ser decisivos en el momento de emitir un juicio o tomar una decisión con respecto al sistema estudiado.

Los costos por defecto que se asignan a los recursos son: costo de ocupación, costo de ociosidad y costo por piezas procesadas o en uso. De éstos, el costo por ociosidad es el único que no se carga a la entidad. A continuación se muestran los reportes correspondientes a los recursos:

Resources



En el explorador de reportes se muestran, para cada réplica, las diferentes opciones de estadísticas para los recursos

Suponga que se desean conocer de manera detallada los costos de funcionamiento de la fresa

Replication 1	Start Time:	0.00	Stop Time:	2,400.00	Time Units: Minutes
----------------------	-------------	------	------------	----------	---------------------

FRESA

	Average	HalfWidth	Minimum	Maximum
Usage				
Number Busy	0.9960	(Correlated)	0	1.0000
Number Scheduled	1.0000	(Insufficient)	1.0000	1.0000
Utilization	0.9960	(Correlated)	0	1.0000
Cost		Value		
Busy Cost	199.20			
Idle Cost	0	Al no tener un costo por tiempo ocioso, esta categoría aparece con 0		
Usage Cost	800.00			
Other		Value		
Number Times Used	400.00			
Scheduled Utilization	0.9960			

El recurso *pulidora*, al tener un costo asociado al tiempo que permanece ocioso, presenta el valor correspondiente en esta categoría.

1. ANIMACIÓN DE ENTIDADES

Consiste en asignarle una figura o dibujo a la entidad que se mueve dentro del modelo.

Para cualquier modelo en general, la figura inicial que adoptan las entidades que ingresan al sistema se define en la hoja de trabajo de la entidad (*Entity*). Si se requiere un posterior cambio de figura, éste se puede realizar mediante el módulo *Assign*.



Entity

En el caso del ejemplo 2.1 resulta preferible tener una figura con forma humana que represente al cliente, en vez del reporte que aparece por defecto. Para esto se cambia la figura inicial (*initial picture*) por una más conveniente; por ejemplo, *picture.woman*.

Hoja de trabajo de la entidad

Entity Basic Process		
	Entity Type	Initial Picture
1	cliente	Picture.Report

Figura inicial por defecto

Se elige, del menú desplegable, la figura más conveniente

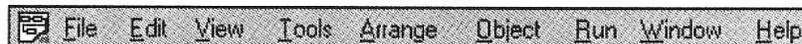
Initial Picture
Picture.Report
Picture.Telephone
Picture.Truck
Picture.Van
Picture.Widgets
Picture.Woman
Picture.Yellow Ball
Picture.Yellow Page

La figura elegida representa a la entidad desde su ingreso al modelo hasta su salida o en un cambio posterior que se haga a la animación de ésta

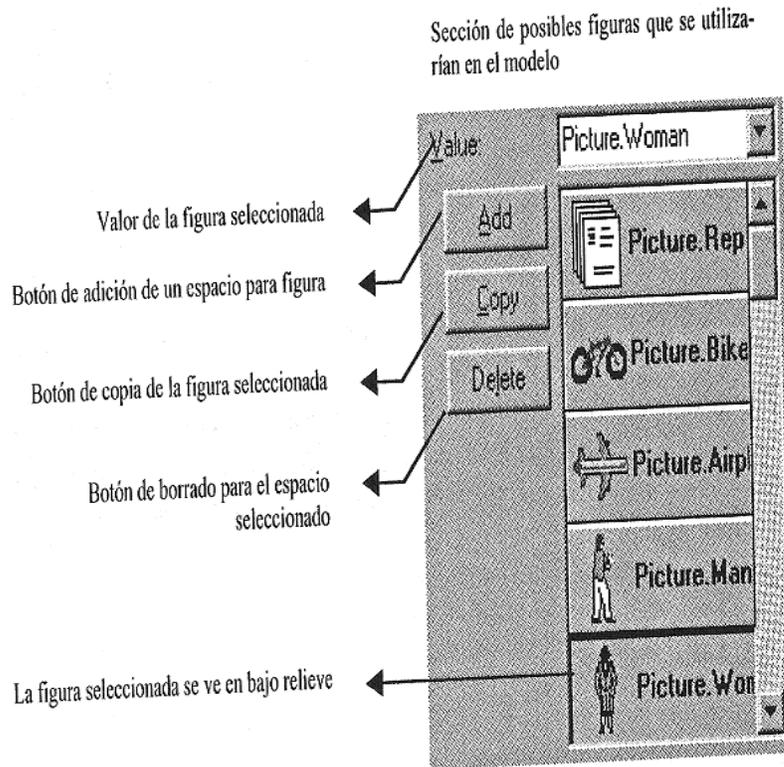
Entity Basic Process		
	Entity Type	Initial Picture
1	cliente	Picture.Woman

Las diferentes figuras que se van a emplear en el modelo se pueden definir y editar en la ventana de reemplazo de figuras (*Entity picture placement*). Se accede a esta ventana desde la barra de menús mediante las instrucciones *Edit/Entity pictures*.

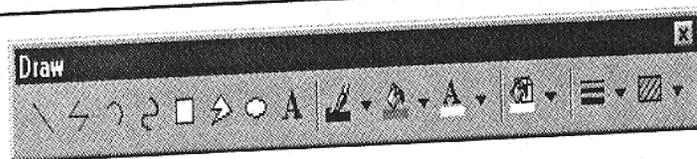
Barra de menús



La ventana de reemplazo de figuras contiene las bibliotecas de dibujos con las que cuenta el programa; también ofrece, mediante la barra de dibujo, la alternativa de crear diseños propios para animaciones.

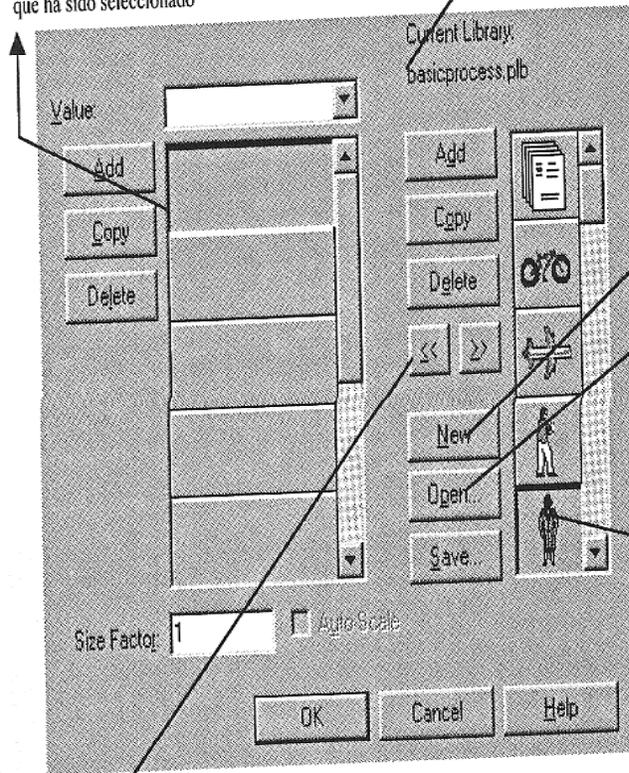


Sugerencia. Se puede modificar una figura existente en la ventana de edición de figuras. Se accede a esta ventana haciendo doble clic en la figura que se desea cambiar. A través de la barra de dibujo se pueden incluir modificaciones, cambiar los colores por medio de la paleta, incluir texto o combinar figuras de diferentes librerías mediante la acción copiar-pegar.



Se pueden extraer figuras de diferentes librerías para usarlas en el modelo. A continuación se describe el procedimiento de esta operación:

2. Se selecciona el espacio en donde se desea ubicar la figura seleccionada de la librería. Cuando se hace clic sobre el espacio, éste queda en bajo relieve, lo que indica que ha sido seleccionado



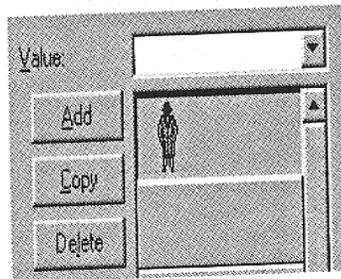
Librería de dibujos actual

Botón para crear nueva librería

Botón para abrir una librería existente. Los archivos de librerías tienen extensión *plb*, y se encuentran en el mismo directorio en el que se ubica el programa

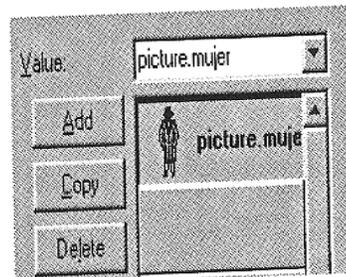
1. Se selecciona la figura deseada de la librería haciendo clic sobre ésta para marcarla. La figura marcada queda en bajo relieve

3. Mediante el botón << se copia la figura de la derecha hacia el espacio seleccionado en la izquierda, como lo indica el sentido de la ilustración del botón. Para el caso inverso, el proceso funciona de la misma manera pero mediante el botón >>



Una vez se ha copiado la figura en el espacio correspondiente, se procede a colocar el nombre o identificación de ésta

La manera correcta de referirse a las figuras en ARENA® es colocar el nombre del objeto (*picture*) seguido por el nombre que define el usuario, separados por un punto. De tal manera que si se quiere identificar la figura elegida como "mujer", se debe llenar el campo en blanco, así como se muestra a continuación:



Picture.mujer



La figura que se acaba de crear se debe asignar en la hoja de trabajo de la entidad como figura inicial, con el fin de que pueda verse en el modelo.



Entity

Entity Type	Initial Picture
cliente	Picture.mujer

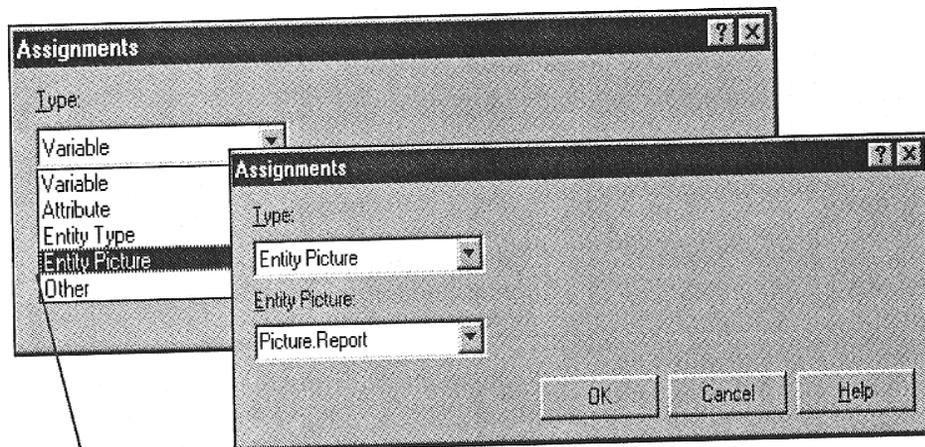
2. CAMBIO EN LA ANIMACIÓN DE ENTIDADES

En ocasiones resulta necesario cambiar la figura de una entidad en determinadas instancias del modelo. Por ejemplo, después de una operación de pintura, las piezas salen con otro color; para esto se emplea el módulo *Assign*.



Assign

En el cuadro de diálogo de este módulo se presenta la opción de cambiar el atributo especial de animación *EntityPicture* por el valor deseado, con su correspondiente sintaxis, teniendo en cuenta que esté definido en la ventana de reemplazo de figuras.

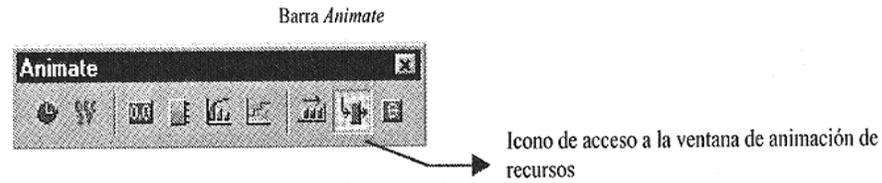


Se escoge el tipo de asignación *EntityPicture*, que es un atributo de uso especial en el cual se almacena la identificación de la figura que representa a la entidad

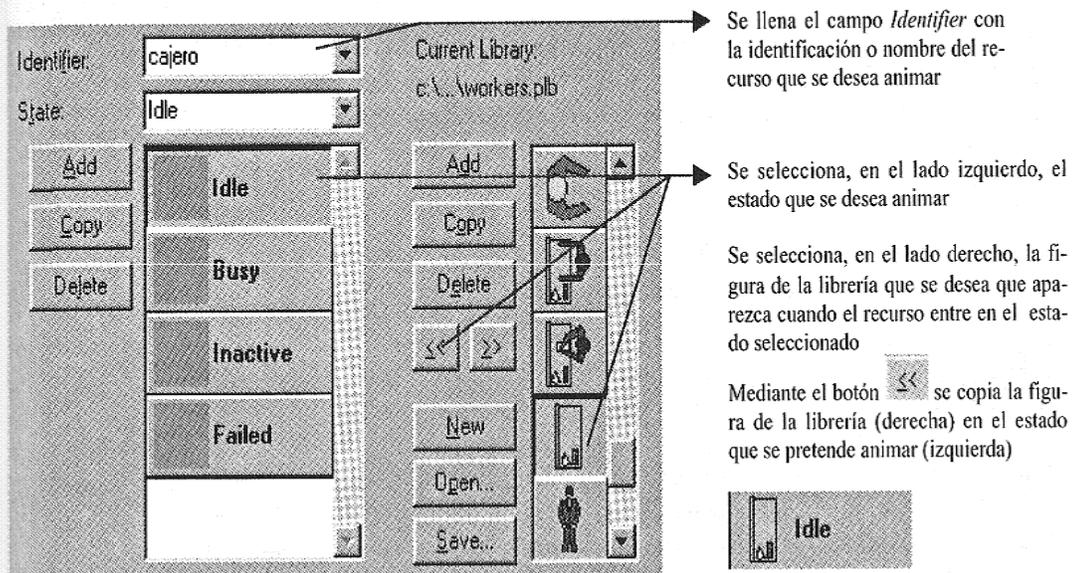
Una vez la entidad ha dejado el módulo, una nueva figura asignada realizará su representación.

3. ANIMACIÓN DE UN RECURSO

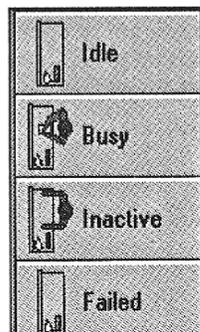
Consiste en asignar una figura específica a cada estado que el recurso pueda adoptar, por ejemplo, ocupado, ocioso, en reparación, etc. Para esto se cuenta con la barra *Animate*, la cual contiene el icono que permite acceder a la ventana de animación de recursos (*Resource picture placement*).



Para la animación de un recurso se debe tener en cuenta la identificación de éste y el estado que se desea animar. Para mayor ilustración a continuación se anima el recurso *cajero* del ejemplo 2.1:



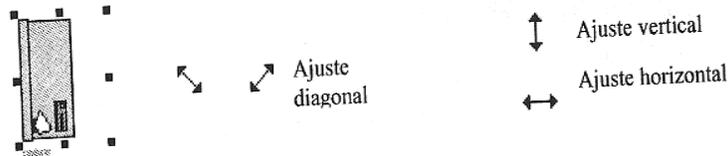
Se escoge, de manera análoga, una figura para cada estado del recurso especificado, lo cual da como resultado una secuencia de movimientos según la situación del recurso.



Los estados por defecto son ocioso (*idle*), ocupado (*busy*), inactivo (*inactive*) y bajo falla (*failed*). Los dos primeros son los estados normales para cualquier recurso; los últimos sólo se muestran si se definen fallas en los recursos o tiempos muertos

Cuando se termina la selección de figuras se presiona el botón , y el puntero del ratón adopta forma de cruz, lo que le permite al usuario especificar el sitio en el cual desea colocar la figura.

Se puede cambiar el tamaño de la figura marcándola (clic simple); así mismo, si se quiere agrandarla o encogerla, se posiciona el puntero del ratón sobre uno de los puntos marcados.



Cuando el puntero adopte la forma de ajuste diagonal, vertical u horizontal, se modifica el tamaño mediante un clic sostenido.

4. ANIMACIÓN DEL VALOR DE UNA VARIABLE EN EL MODELO

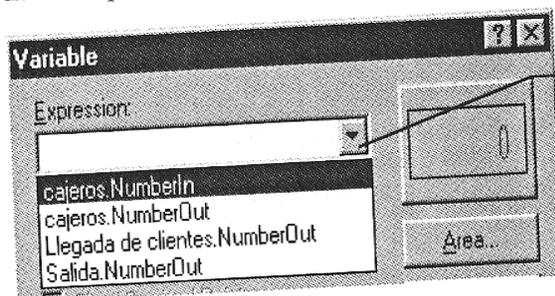
Consiste en mostrar de manera numérica o gráfica el valor actual en tiempo simulado de una variable, con el fin de verificar el funcionamiento de la lógica empleada, estudiar su comportamiento en el tiempo o, simplemente, por razones estéticas.

VARIABLE

Para mostrar el valor numérico de una variable, Arena® cuenta con una herramienta gráfica llamada *Variable*, que se encuentra en la barra *Animate*.

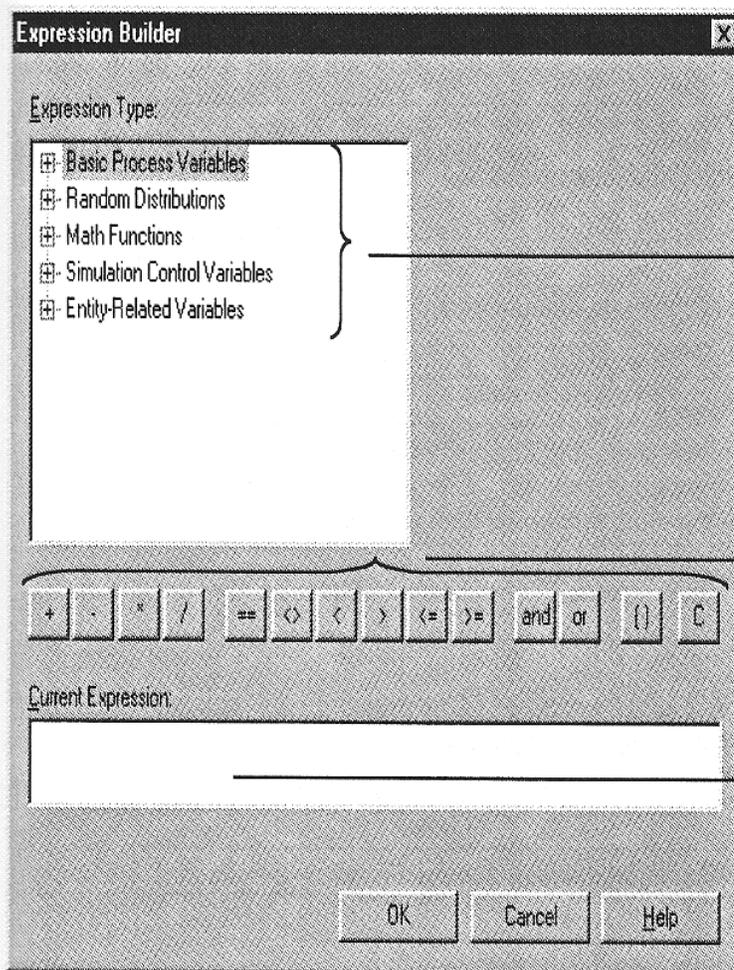
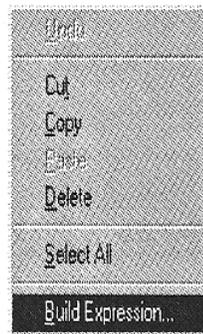


Cuando se activa esta función se muestra un formulario que contiene un campo llamado *Expression*, en el cual se coloca el nombre de la variable que se desea animar. Este campo cuenta con un menú desplegable que contiene las variables predeterminadas por el programa, como *información sobre procesos* o *entradas a módulos*. Así, para el ejemplo 2.1 del capítulo 2 es (Banco) *cajeros.NumberIn* (entradas al módulo *Process* llamado cajeros). Además de estas variables predeterminadas, se encuentran listadas aquellas que define el usuario. Ejemplo, *inventario*.



Menú desplegable con los nombres de las variables que predetermina el programa para los módulos incluidos

Es posible que la variable que se desea animar no se encuentre en la lista, pues se necesita que ésta se construya mediante una expresión. Con el fin de hacer esta tarea más sencilla, ARENA® posee un asistente llamado *Expression builder*. Se accesa a esta herramienta haciendo clic derecho en los campos que sean susceptibles de ser llenados con una expresión. Ejemplo, el campo *Expression* en un módulo *Decide*, o *Condition* en un módulo *Hold*, configurado como *Scan for condition*.



Fuentes de las cuales procede la expresión que se desea construir

Pueden desplegarse las diferentes opciones que ofrece cada categoría haciendo clic en 

Operadores matemáticos y lógicos en forma de botones. Cuando se les presiona, estos operadores se incluyen dentro de la expresión que se construye

Expresión actual. Muestra la expresión que se está construyendo

Basic process variables

Estas variables se relacionan con los siguientes elementos:

Elemento	Ejemplos
Entidades (<i>Entities</i>)	Costos, número de entidades que entran, número de entidades que salen, etc.
Procesos (<i>Process</i>)	Costos, número que entra al módulo, número que sale, WIP, etc.
Colas (<i>Queue</i>)	Número en cola, tiempo promedio de espera, suma de atributos de entidades en cola, etc.
Recursos (<i>Resource</i>)	Costos, estado (ocupado, ocioso, falla) o uso (porcentajes de tiempo ocioso), capacidad activa, etc.
Programación (<i>Schedule</i>)	Valor actual
Grupos (<i>Set</i>)	Número de miembros del grupo, posición de un miembro en el grupo, recursos ocupados dentro de un set, etc.
Variables definidas por el usuario (<i>Variable</i>)	Valor actual, valor mínimo y máximo y valor promedio.

Advanced process variables

Estas variables se relacionan con los siguientes elementos:

Elemento	Ejemplos
<i>Advanced set</i>	Número dentro del grupo, número de miembros del grupo
<i>Expression</i>	Valor actual
<i>Stateset</i>	Valor del estado
<i>Statistics</i>	<i>Counter</i> (valor actual, límite), <i>Frequency</i> (número de ocurrencia, tiempo promedio en categoría, etc.), <i>Tally</i> (número de observaciones, valor promedio, etc.), <i>Time-persistent</i> (mínimo, máximo, promedio, etc.).
<i>Storage</i>	Número en almacenamiento

Advanced transfer variables

Estas variables se relacionan con los siguientes elementos:

Elemento	Ejemplos
<i>Conveyor</i>	Estado, longitud, velocidad, etc.
<i>Distance</i>	Distancia entre estaciones
<i>Sequence</i>	Estación dentro de la secuencia
<i>Transporter</i>	Estado, velocidad, número de unidades ocupadas, etc.

Random distributions

En esta categoría se dispone de ayudas para la construcción de expresiones relacionadas con distribuciones de probabilidad; por ejemplo, para la distribución normal el asistente muestra los campos en los que serán ingresadas la media y la desviación estándar.

Math functions

En esta categoría se encuentran las funciones matemáticas definidas para ARENA[®]. Por ejemplo, raíz cuadrada (SQRT), logaritmo natural (LN), parte entera (AINT), resto de la división entera (MOD), etc.

Simulation control variables

Están relacionadas con la corrida del modelo:

Variable	Uso
Máximo número de réplicas (MREP)	Número de corridas del modelo que se ejecutarán. Es asignable por el usuario*.
Número actual de réplicas (NREP)	Número de la réplica del modelo que se está corriendo. No es asignable por el usuario.
Tiempo actual de simulación (TNOW)	Tiempo que ha transcurrido desde el inicio de la corrida. No es asignable por el usuario.
Tiempo de finalización de simulación (TFIN)	Tiempo en que finalizará la corrida. Es asignable por el usuario*.

* El valor de estas variables se puede modificar en tiempo de diseño en el menú *Run/Setup/Replications Parameters* o en tiempo de ejecución, mediante un módulo *Assign*.

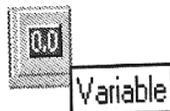
Entity-Related variables

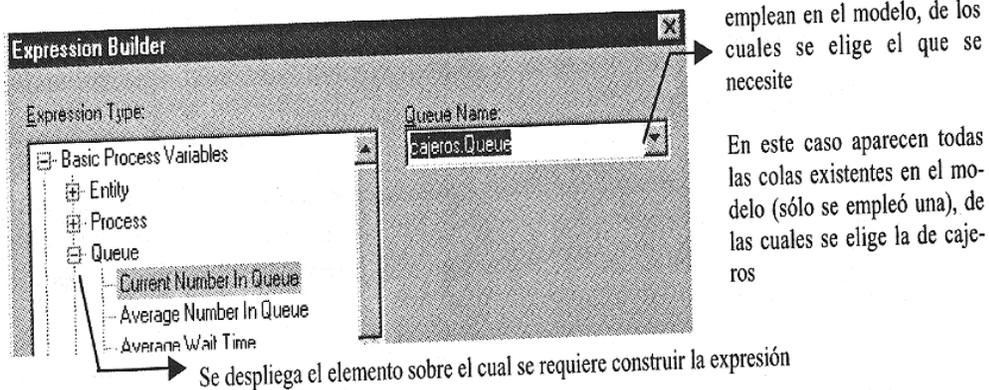
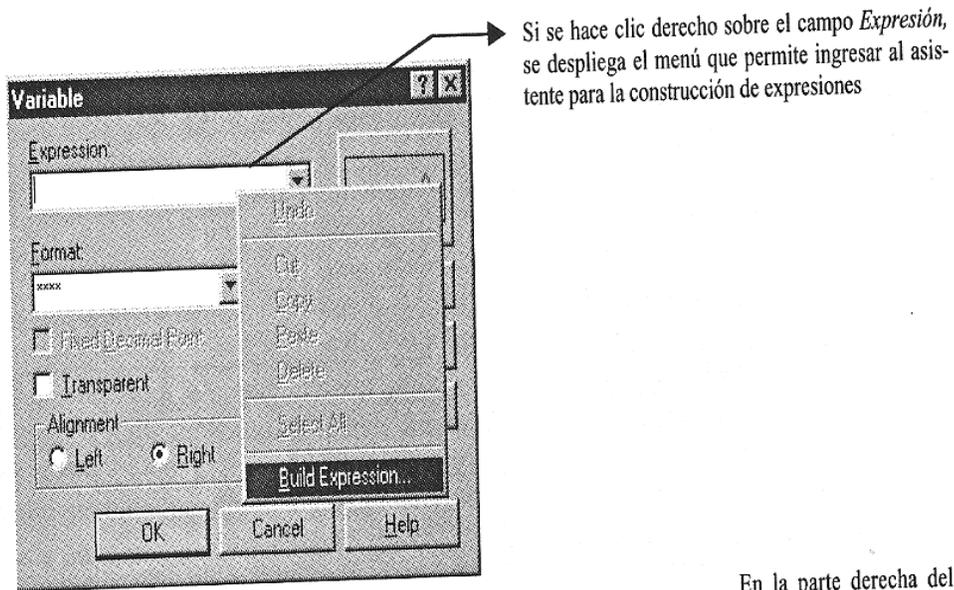
Se refiere a las variables que están relacionadas con la entidad, como atributos, costos, tiempos y atributos definidos por el usuario. También se incluyen las relacionadas con los grupos o lotes (entidades representativas de un grupo). (Ver *Formación y desmembración de lotes o grupos*, capítulo 2).

Elemento	Ejemplos
Atributo	Tipo de entidad, número de serie, secuencia, paso dentro de la secuencia.
Variables de grupos	Número de entidades en el grupo, suma de atributos, etc.

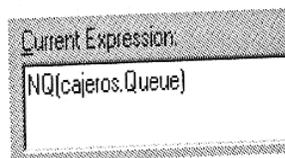
Suponga que en el ejemplo 2.5 se desea establecer el número de clientes que se encuentran en la cola del cajero en todo momento, con el fin de verificar si la lógica del modelo se está ejecutando correctamente.

Para esto se coloca un cuadro de variables, dentro del cual se construye la expresión correspondiente:





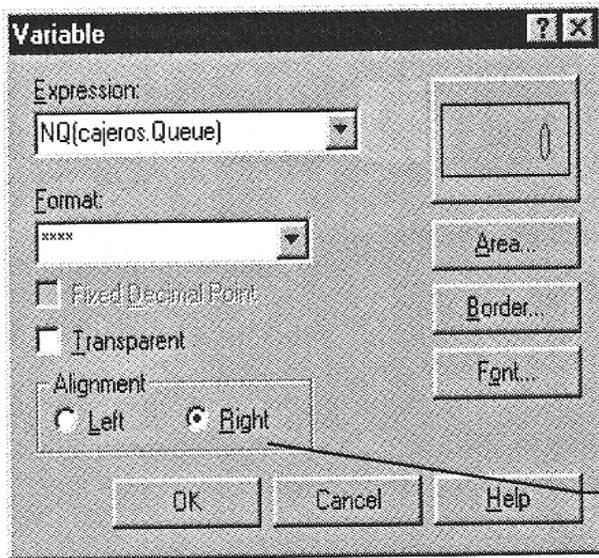
En este caso se buscan las expresiones relacionadas con las colas dentro de las variables de procesos básicos. Una vez allí, se selecciona la información requerida, que, en este caso, es el número actual en cola (*Current number in Queue*)



Al tiempo que se selecciona la información requerida, se construye la expresión en el campo marcado como *Current Expression*

Según las necesidades del modelador, se pueden hacer combinaciones de expresiones matemáticas, operadores lógicos, variables de procesos, transportes y distribuciones.

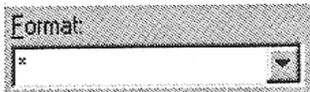
Cuando se concluye, se presiona el botón *ok* y se retorna al formulario inicial, el cual se muestra a continuación:



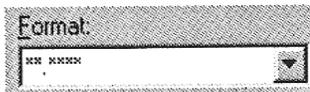
Formato de cuadro de *Display*, como color de área y bordes; además, tamaño estilo y color de fuente

Alineación del número en el cuadro de *Display*

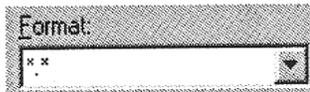
El campo *Format* se refiere al formato del número que se va a mostrar en el *Display*; viene dado por las posiciones decimales y enteras que se requieran más el signo, en el caso de que se trate de valores negativos. Estas posiciones se indican mediante asteriscos o signos de multiplicación, igual a los que se encuentran en el teclado numérico del ordenador (*). El usuario puede ingresar estas posiciones; aunque para facilitar esta labor en algunos casos se cuenta con formatos predeterminados.



Número de un dígito entero positivo, pues no hay espacio para el signo. Para expresar un número de un dígito que pueda ser negativo se utiliza **

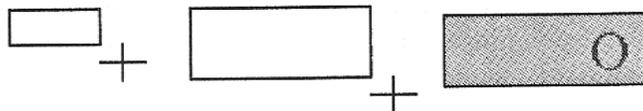


Número de dos dígitos enteros y cuatro decimales si es positivo, y un entero y cuatro decimales si es negativo, ya que una posición es para el signo



Número de un dígito entero y un dígito decimal si es positivo (sin signo). Cuando esta posición toma valores negativos, sólo admite un dígito decimal

Cuando se concluye la edición del formulario del *Display*, el cursor pasa a ser una cruz, lo cual indica que se está en modo de dibujo, y si se hace clic sostenido, se dibuja un rectángulo (*Display*).



En el recuadro construido se visualiza, en todo momento, el número en cola de la cola de los cajeros, tal como se muestra en la ventana anterior, en *Expression* *NQ(cajeros.Queue)*.

5. GRÁFICAS

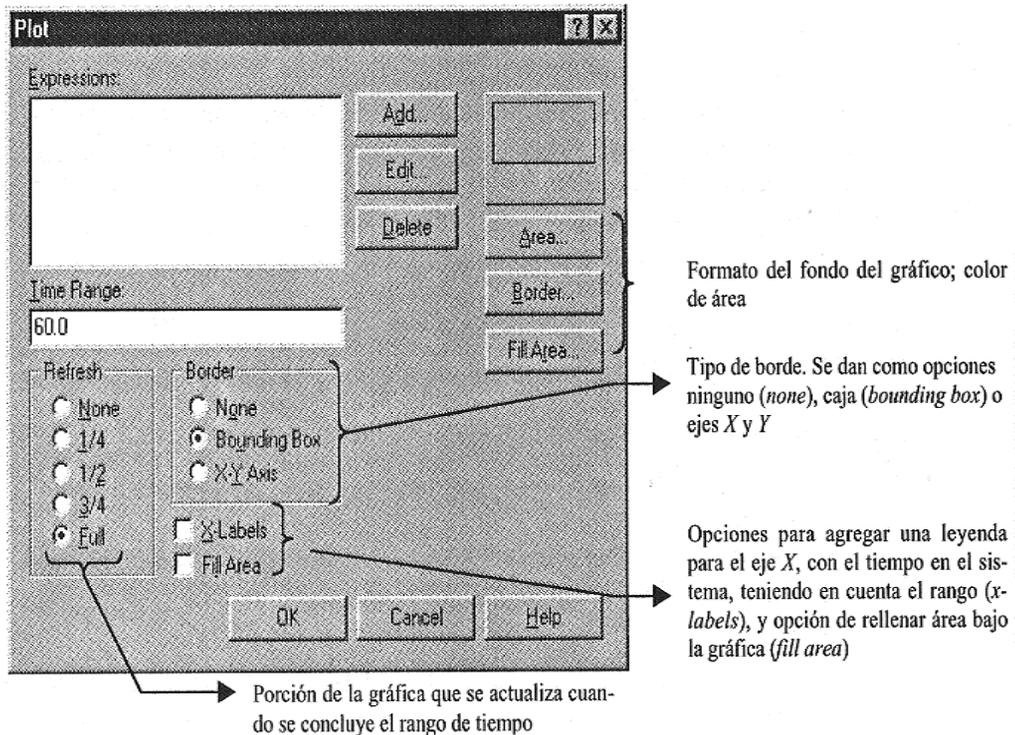
OPCIÓN PLOT

Es un recurso de animación de variables en forma de gráfico en un plano coordenado cuyo eje X es el tiempo y el eje Y , el valor de la variable. En otras palabras, se trata de un recurso que grafica el valor de la variable de interés en el tiempo.

Se accede a esta herramienta mediante el icono de *Plot* en la barra *Animate*.



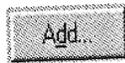
El formulario de la función *Plot* es un poco más complejo que el de la anterior, puesto que permite graficar varias expresiones o variables al mismo tiempo.



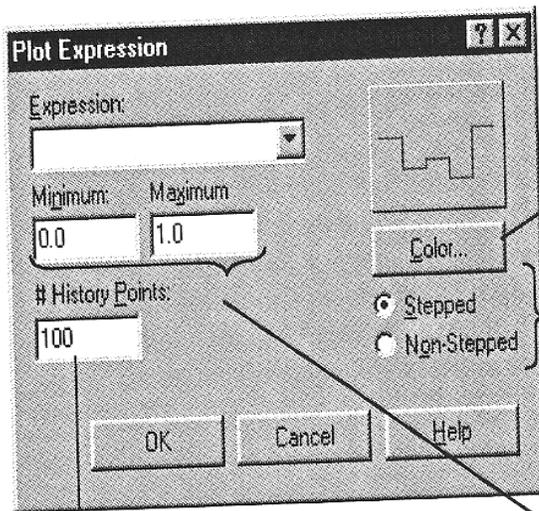
El campo *Time Range* se refiere al intervalo de tiempo en el cual se desea graficar el valor de la variable. Por ejemplo, si se desea ver el comportamiento de una variable cada hora, el valor de este campo será 60 minutos; ahora, si se desea observar un día de 8 horas, el valor será 480 minutos.

Cuando se finalice el rango de tiempo, por ejemplo, al llegar a los primeros 60 minutos, la gráfica se puede actualizar completamente o sólo la mitad ($1/2$), así como cualquier cantidad de las gráficas contenidas en las opciones del cuadro *Refresh*.

Las expresiones que se van a graficar se agregan mediante el botón *Add*.



El formulario para ingresar la expresión correspondiente es similar al que se observa en la sección de *Display*, ya que también cuenta con el asistente para la construcción de expresiones.



Color de la línea del gráfico

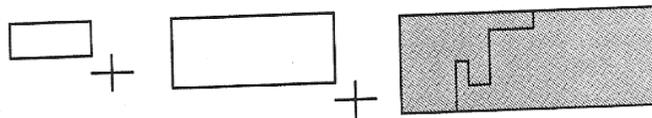
Tipo de gráfico; puede ser escalonado o no escalonado. (Escalonado: baja o sube en línea recta desde el valor anterior)

Valores mínimo y máximo que la variable puede asumir; esto determina la escala para la expresión actual. Aunque se pueden incluir diferentes escalas para las expresiones que se grafiquen, esto no es recomendable para efectos de comparación

Cantidad de puntos que se desea mantener en la gráfica. Un valor de 100 (predeterminado) significa que cuando se llega a esta cantidad de puntos graficados, se elimina el último al incluir uno nuevo para mantener este valor. Como consecuencia, la gráfica tendrá el efecto visual de desplazarse en el rango

Siguiendo con el mismo ejemplo, mediante el uso del asistente se construye la expresión que indica el número en cola en el recurso cajeros: $NQ(\text{cajeros.Queue})$

Cuando se editan todos los formularios y se acepta, el puntero del ratón se transforma en cruz; entonces, se procede a construir el cuadro que contendrá la gráfica.



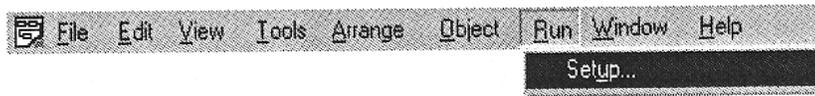
En el recuadro construido se visualiza en todo momento la gráfica del número en cola de la cola de los cajeros en el ejemplo analizado.

GENERACIÓN Y ANÁLISIS DE REPORTES

1. CONTROL DE CORRIDAS

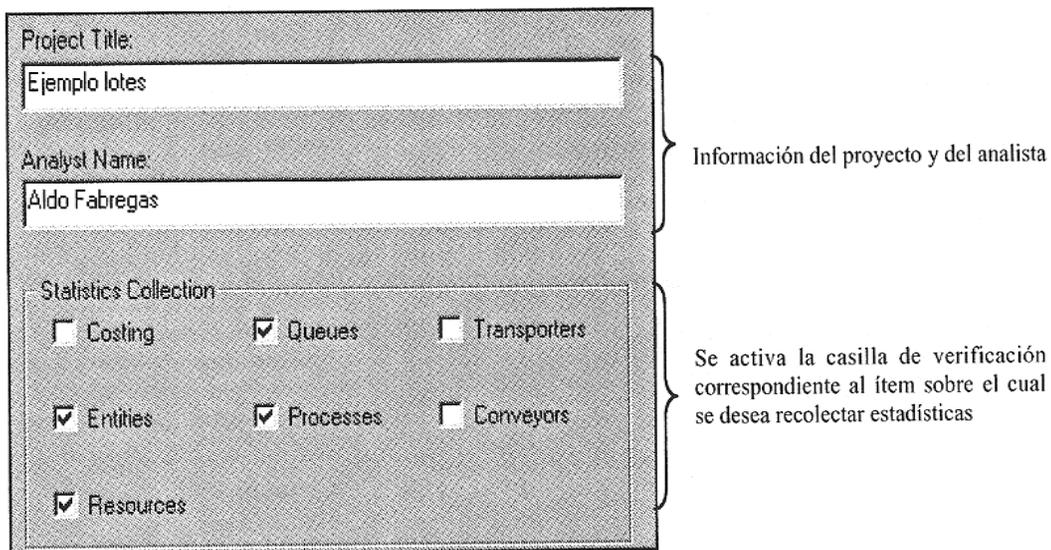
Luego de construir el modelo de simulación se deben fijar los parámetros necesarios para la corrida de éste. En ARENA® se pueden incluir, como parámetros del proyecto, y según el modelo, la información del modelo, el nombre del analista y el tipo de estadísticas que se va a recolectar. Por otra parte, el número de réplicas, el período de estabilización, la longitud de la corrida y las unidades básicas de tiempo, entre otros aspectos, se incluyen como parámetros de la corrida.

El control de corridas se hace desde la barra de menús y mediante las instrucciones *Run/Setup*.



Suponga que se desea analizar el ejemplo 2.6, *Batch-Separate*. (Ver *Formación y desmembración de un lote de transferencia*, capítulo 2).

Para ingresar al cuadro de diálogo *Project Parameters* se hace clic en la pestaña de parámetros del proyecto.



Si no se va a trabajar con costos, se recomienda de manera especial desactivar la casilla de recolección de estadísticas relacionada, ya que de lo contrario se recarga el reporte y, así, se hace engorrosa su interpretación.

Luego, si se hace clic en la ventana de parámetros de la corrida, se accede al siguiente cuadro de diálogo: *Replication Parameters*.

Número de corridas o réplicas

Iniciación de estadísticas. Si no se activa, en el momento de calcular las estadísticas se mezclan datos de todas las corridas

Opción de inicialización del sistema entre réplicas. Si no se activa, cada réplica no comenzaría con 0 entidades, sino que partiría de las condiciones finales de la anterior

Período de estabilización y unidades de tiempo en las que éste viene expresado

Longitud de la corrida y unidades de tiempo en las que éste viene expresado

Duración de un día de trabajo en horas; puede tomarse como jornada laboral. El campo *Base time units* expresa las unidades de tiempo en las cuales se basa el modelo tanto para la visualización como para los reportes

Condición para la finalización de la corrida, por ejemplo, cuando se ha terminado cierto número de piezas de un tipo

Los reportes se constituyen también en un aspecto de interés. Mediante la pestaña *Reports* se accede a su cuadro de diálogo, en el cual se define el reporte predeterminado y la manera como éste aparecerá:

Reports

At the end of the simulation run, display default report...

Always Never Prompt me

Modo de mostrar el reporte predeterminado al finalizar la simulación. Las opciones que se pueden escoger son nunca (*never*), siempre (*always*) y, la más recomendada, preguntar antes de mostrar (*prompt me*)

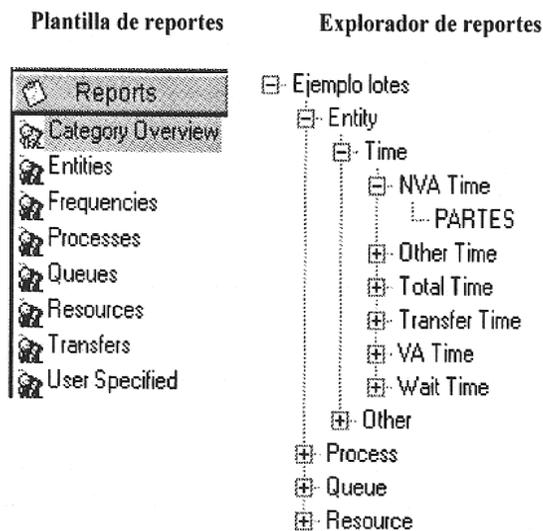
A continuación se muestra el menú desplegable, el cual permite seleccionar el tipo de *Report* que se desea como predeterminado:

Opciones
<i>Category overview</i>
<i>Category by replication</i>
<i>Entities</i>
<i>Frequencies</i>
<i>Processes</i>
<i>Queues</i>
<i>Resources</i>
<i>Transfer</i>
<i>User specified</i>
<i>SIMAN summary report (.out file)</i>

El *SIMAN summary report* se genera en un archivo de texto, y para su creación no se necesita de una base de datos. Por esta razón, cuando se fija como predeterminado se visualiza una casilla de verificación en la que se hace posible deshabilitar la generación de la base de datos para el reporte.

2. MANEJO DE REPORTES

El reporte que aparece es el que se señala como predeterminado. Si se escoje el resumen general (*Category overview*), la siguiente será la primera hoja que se verá:

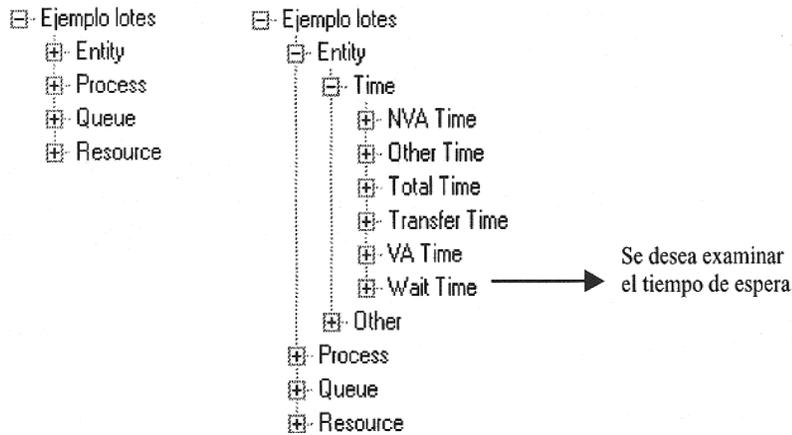


Se puede ir de una categoría de reporte a otra mediante la plantilla de reportes que se encuentra en la barra de proyectos. Dentro de cada categoría se pasa de un ítem a otro por medio del explorador de reportes, de manera análoga a como se maneja el explorador de WINDOWS.

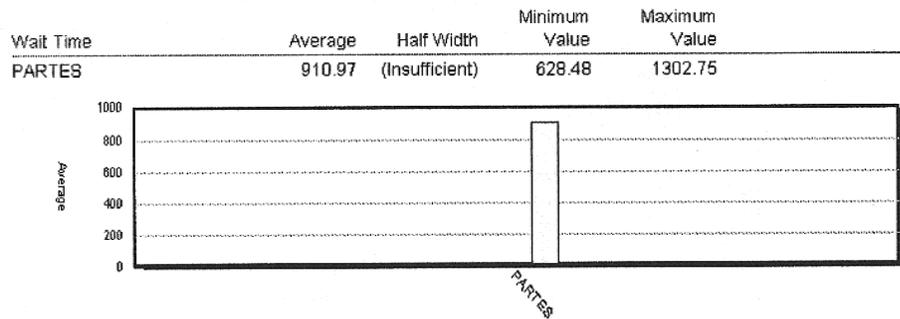
El primer reporte que encabeza la lista es el resumen general (*Category overview*). Éste contiene toda la información de entidades, recursos, procesos, transferencias, etc. La información se muestra en forma numérica y gráfica; así mismo, se visualizan los promedios y las desviaciones de los diferentes tiempos analizados.

Se puede acceder a las diferentes secciones del reporte por medio del explorador.

Para visualizar los informes referentes a las entidades, se despliega la sección correspondiente y se elige el ítem específico que se desea observar:



Suponga que de la categoría entidades se desea examinar el tiempo de espera:



De forma parecida se consulta cada categoría dentro del resumen general (*Category overview*).

Importante. En los casos en los que se presentan varias réplicas, el resumen general muestra los datos promediados de todas las corridas o réplicas; si se desean ver para cada réplica los reportes por categoría, se debe recurrir a los reportes individuales en la plantilla de reportes.

Si se desea ver sin gráficos información sobre las entidades, se debe seleccionar esa categoría dentro de la plantilla de reportes; aparece, entonces, la primera hoja del reporte, que contiene un resumen de la información para todos los tipos de entidades:

Entities

Informe de entidades entrantes vs. entidades salientes

Other

	Number In	Number Out
PARTES	544	529
Total	544	529

El resumen de entidades se presenta para cada réplica realizada.
El encabezado muestra el tiempo de inicio, el tiempo de finalización y las unidades de tiempo base para el informe

08:51:00a.m.

Entities

Enero 9, 2000

Ejemplo lotes

Replications: 1

Replication 1

Start Time: 0.00 Stop Time: 2,400.00 Time Units: Minutes

Entity Detail Summary

Éstos son los valores promedios para las diferentes categorías de tiempo.
Estas categorías se deben especificar dentro del modelo; adquieren más relevancia cuando se tabajan costos

Time

	VA Time	NVA Time	Transfer Time	Wait Time	Other Time	Total Time
PARTES	285.02	0.00	0.00	910.97	0.00	149.25
Total	285.02	0.00	0.00	910.97	0.00	149.25

	Tiempo de valor agregado	Tiempo de no valor agregado	Tiempo de transferencias o transportes	Tiempo de esperas	Otros tiempos	Tiempo total o en el sistema
PARTES	285.02	0.00	0.00	910.97	0.00	149.25
Total	285.02	0.00	0.00	910.97	0.00	149.25

Mediante el explorador de reportes se pueden visualizar reportes más detallados (por tipo de entidad) al desplegar las subcategorías posibles:

Otras estadísticas

Other		Value			
Number In	Entidades entrantes	544			
Number Out	Entidades salientes	529			
WIP	Trabajo en proceso	23.0322	(Correlated)	0.00	42.0000

La abreviatura WIP corresponde a *work in process* (trabajo en proceso); su promedio se puede tomar como el número de entidades en el sistema

- [-] Ejemplo lotes Nombre del proyecto
- [-] Replication 1 Número de réplica
- [-] PARTES Tipo de entidad (*EntityType*)
- [-] Time Resultados de tiempos
- [-] Other Otros resultados

08:51:00a.m.

Entities

Enero 9, 2000

Ejemplo lotes

Replications: 1

Replication 1

Start Time: 0.00 Stop Time: 2,400.00 Time Units: Minutes

A través del explorador se acceden a los reportes que contienen análisis detallados acerca de las entidades

Tipo de entidad	Promedio	Mitad del intervalo de confianza	Mínimo	Máximo
PARTES				
Time	Average	Half Width	Minimum	Maximum
NVA Time	0.00	(Insufficient)	0.00	0.00
Other Time	0.00	(Insufficient)	0.00	0.00
Total Time	140.25	(Insufficient)	109.23	181.88
Transfer Time	0.00	(Insufficient)	0.00	0.00
VA Time	265.02	(Insufficient)	259.90	271.02
Wait Time	910.97	(Insufficient)	628.48	1,302.75

Estos resultados pueden ser interpretados como: *promedio ± mitad del intervalo de confianza (average ± half width)*

El término *insufficient* significa que no se alcanzó un número de muestras óptimo (por lo menos 320) para calcular un intervalo de confianza. El término *correlated* significa que las muestras no son independientes entre sí y que es necesario buscar una longitud de corrida para aquellas que no se correlacionen.

3. INTERPRETACIÓN DE RESULTADOS

A continuación se explica el contenido de cada reporte; además, se dan las pautas que permiten su correcta interpretación para las funciones del panel de procesos básicos.

3.1 ENTIDADES (*ENTITIES*)

Este reporte consta de dos secciones:

a. *Estadísticas de tiempo*

Contienen información sobre la media, el intervalo de confianza y los valores mínimos y máximos con respecto a las diferentes categorías de tiempo. Estas categorías pueden ser:

- Tiempo de valor agregado
- Tiempo de no valor agregado
- Tiempo de espera
- Tiempo de transferencia
- Otros tiempos

La asignación de los tiempos de las diferentes actividades a cada una de estas categorías se debe hacer durante la fase de construcción del modelo. Cada acción que implique una demora se debe analizar previamente y asignar a alguna de las categorías existentes.

b. *Otras estadísticas*

Esta sección contiene el número de entidades que ingresan al sistema (*NumberIn*), el número de entidades que salen del sistema (*NumberOut*) y el número que se encuentra dentro del sistema o trabajo en proceso (WIP) por cada tipo de entidad (*EntityType*).

3.2 FRECUENCIAS (*FREQUENCIES*)

Reportan la frecuencia en el tiempo de la ocurrencia de eventos, como el estado de un recurso o aquel en el cual una expresión adopta valores dentro de unos límites que especifica el usuario.

Las frecuencias se calculan y reportan para cada réplica; éstas no aparecen en el resumen general, ya que en éste se promedian valores de todas las corridas.

Este reporte presenta la siguiente información:

- *Number obs.* Número de veces en que la expresión igualó o estuvo dentro del intervalo especificado, o número de veces en que el recurso estuvo en el estado de interés.
- *Average time.* Tiempo promedio en que la expresión estuvo en el rango o el valor determinado, o tiempo promedio en que el recurso estuvo en un estado específico.

- *Standard percent*. Porcentaje promedio del tiempo en que la expresión estuvo en el rango o el valor determinado, o porcentaje promedio del tiempo en que el recurso estuvo en un estado específico.
- *Restricted percent*. Porcentaje restringido del tiempo en que la expresión estuvo en el rango o el valor determinado, o porcentaje restringido del tiempo en que el recurso estuvo en un estado específico.

3.3 PROCESOS (*PROCESES*)

Este reporte recolecta estadísticas de las actividades que se desarrollan en los módulos *Process*, ya sean *standard* o *submodelos*. En un determinado reporte, sin importar que utilice o no un recurso, habrá estadísticas por cada módulo con el que cuente el modelo.

Contempla tres secciones, a saber:

a. *Tiempo (time)*

En esta sección se proveen estadísticas, como media, intervalo de confianza y valores máximo y mínimo, para la categoría de tiempo que la(s) actividad(es) del módulo *Process* tenga asignada (valor agregado, no valor agregado, etc.). Si se tiene un recurso asociado, se incluyen estadísticas de tiempo de espera. Se interpreta como el tiempo que pasa una entidad en las actividades desarrolladas en el módulo *Process* bajo la categoría de tiempo que se le asigna a cada una. En el caso de submodelos, se puede tener múltiples actividades en diferentes categorías de tiempo.

b. *Tiempo acumulado (accumulated time)*

Es el tiempo acumulado por todas las entidades que pasan por el módulo *Process* para cada una de las categorías de tiempo.

c. *Otras estadísticas (other)*

En esta sección se muestra el número de entidades que entraron al módulo (*NumberIn*) y el número de entidades que salieron (*NumberOut*).

3.4 COLAS (*QUEUES*)

Este reporte muestra la recolección de estadísticas de tiempo y número de entidades para todas las esperas registradas en el modelo. Se divide en dos secciones que corresponden a cada espera:

a. *Tiempo (time)*

En esta sección se proveen estadísticas, como media, intervalo de confianza y valores máximo y mínimo, para el tiempo de espera en una cola determinada.

b. *Otras estadísticas (other)*

En esta sección se proveen estadísticas, como media, intervalo de confianza y valores máximo y mínimo, para el número de entidades en espera (*longitud de cola*) en una fila determinada.

3.5 RECURSOS (*RESOURCES*)

En este reporte se recolectan estadísticas que usan los diferentes recursos empleados en el modelo. Se divide en dos secciones, que corresponden a cada recurso:

a. *Uso (usage)*

En esta sección se proveen estadísticas, como media, intervalo de confianza y valores máximo y mínimo, para el número de unidades de recurso ocupadas (*Number busy*), el número de unidades programadas (*Number scheduled*) y la utilización. Estas estadísticas adquieren mayor relevancia cuando la capacidad de un recurso es variable.

Number busy. Número de unidades de recurso que se ocuparon. Esto resulta de particular interés cuando un recurso tiene la capacidad para atender a más de una entidad a la vez.

Number scheduled. Número de unidades de recurso programadas, es decir, se refiere al número máximo de entidades que un recurso de capacidad fija puede atender, o al promedio de atención en el caso de programación de la capacidad.

b. *Otras estadísticas (other)*

En esta sección se encuentran los siguiente ítems:

Number times used. Número de veces en que una entidad toma control del recurso.

Scheduled utilization. Utilización programada del recurso. Se refiere a la utilización sólo en el período de tiempo para el cual el recurso está realmente programado en el sistema.

PANEL DE PROCESOS AVANZADOS

El panel de procesos avanzados es un conjunto de módulos de funciones especializadas que dan un mayor grado de detalle y control al flujo de entidades en un modelo. En esta plantilla de modelamiento se pueden manejar archivos externos (lectura y escritura) al igual que hojas de cálculo o archivos de texto; así mismo, se pueden generar estadísticas y, en general, complementar funciones del panel de procesos básicos.

A continuación se presentan los módulos que componen el Panel de procesos avanzados con una breve descripción de su función:

1. MÓDULOS LÓGICOS

Delay. Este módulo se encarga de retener las entidades que lleguen a él por el tiempo especificado (puede ser una distribución). Retiene las entidades conforme entran al módulo, y puede demorar o retrasar a varias de ellas simultáneamente. Las entidades se liberan a medida que van cumpliendo su tiempo de demora.

Dropoff. Se encarga de retirar una cantidad específica de entidades de un grupo previamente formado (módulo *Pick up*) a partir de una posición determinada. Una entidad representativa portadora de un grupo de entidades en el cual cada entidad tiene un posición definida (tal como en una fila) puede ingresar a este módulo y dejar una parte o el total de su carga (entidades) según las condiciones que especifique el analista.

Hold. Cumple tres funciones que afectan el flujo de entidades en el modelo. La primera es retener entidades hasta que una señal (enviada por un módulo *Signal*) dé la orden de liberar todas o una parte de éstas. La segunda función es retener entidades y, con base en la evaluación de una condición del sistema, liberarlas sólo cuando esta condición sea verdadera. La tercera función de este módulo consiste en retener entidades indefinidamente; es decir, se trata de una espera infinita a través de la cual las entidades permanecen en una fila hasta que sean removidas por alguna lógica externa.

Match. Sincroniza dos o más entidades; es decir, retiene varios tipos de entidades en filas independientes y libera, al mismo tiempo y si las hay, una de cada tipo. En caso de que falte un tipo de entidad, el resto espera a que una de ese tipo llegue y sólo así una de cada tipo se liberará a la vez. Este módulo es de particular utilidad en ensambles.

Pick up. En este módulo una entidad (portadora) recoge otras entidades de una cola especificando la cantidad y la posición en la cola de la cual se van a recoger. Las entidades retiradas forman un grupo; la entidad portadora o representativa de este grupo lo

carga hacia otras instancias lógicas y, en otros casos, físicas del modelo. Las entidades del grupo se descargan mediante un módulo *Dropoff*.

ReadWrite. Este módulo permite leer datos de un archivo externo y trasladar su valor a variables que se usan en el modelo. También permite escribir valores de atributos o variables en archivos externos, con el propósito de que sean analizados en otros programas.

Release. Se encarga de liberar un recurso. Esta función está contenida dentro de un módulo *Process* (Panel de procesos básicos), pero es de gran utilidad cuando se desea un mejor control sobre la lógica que se refiere a los recursos.

Remove. Su objetivo es retirar una entidad que ocupa una posición específica dentro de una cola determinada. Este módulo es activado por una entidad lógica (la que retira), y su salida la conforman, por un lado, esta entidad lógica y, por otro lado, la entidad removida o retirada de la cola.

Seize. Mediante este módulo una entidad toma control u ocupa un recurso determinado. Esta función sólo se encarga de reservar el recurso para la entidad que lo está solicitando, pero no efectúa ninguna demora con éste. En caso de que la entidad no pueda tener acceso al recurso, porque éste se encuentra ocupado, esperará en una fila hasta que esté disponible. El Panel de procesos avanzados cuenta con módulos para el control detallado de las operaciones con recursos; estos modelos son: *Seize*, *Delay* y *Release* (ocupa, demora y libera el recurso).

Search. Se encarga de buscar una condición específica en un lote, fila o grupo de entidades. Este módulo devuelve, en una variable llamada *J*, la posición dentro del lote, fila o grupo de la entidad que cumple con la condición especificada. Esta variable adopta valor 0 cuando no encuentra ningún elemento que cumpla con la condición deseada. El módulo *Search* tiene dos salidas excluyentes; una para cuando la entidad que realiza la búsqueda encuentra una posición que cumple con la condición, y otra para cuando no encuentra elementos que la satisfagan.

Signal. Este módulo se encarga de enviar una señal de un tipo específico a todos los módulos *Wait* que estén esperando esta señal para liberar entidades que se encuentren retenidas. La señal se envía cuando una entidad pasa a través del módulo *Signal*.

Store. Determina el inicio de un almacenamiento con fines de animación y control. Es decir, con este módulo se fija la animación de una entidad en un lugar específico mientras realiza operaciones lógicas. La animación se libera mediante el módulo *Unstore*.

Unstore. Se usa para liberar la animación de una entidad cuando ésta se ha fijado o almacenado en un lugar mediante un módulo *Store*. Entre un módulo *Store* y un *Unstore* se pueden colocar cualquier cantidad de módulos; sin embargo, la entidad sólo se ve en el sitio que define el módulo *Store* hasta cuando se active el módulo *Unstore*.

2. MÓDULOS DE DATOS

Advanced set. Al igual que el módulo *Set* del Panel de procesos básicos, el *Advanced set* permite tomar grupos o arreglos de elementos, en este caso más avanzados, como grupos de colas o secuencias y, en general, de cualquier tipo.

Expression. Almacena expresiones que pueden usarse o llamarse desde cualquier parte del modelo. La utilidad de las expresiones radica en que éstas pueden contener operaciones (suma, producto, etc.) o distribuciones de probabilidad (normal, exponencial, etc.).

Failure. En este módulo se definen las características de una falla o parada de un recurso. Se especifica el tipo de falla (por conteo o por tiempo), el tiempo de reparación, las unidades del tipo de reparación, etc. Estas fallas se referencian en el módulo de datos del recurso que se encuentra en el Panel de procesos básicos.

File. En este módulo de datos se definen los archivos con los que se va trabajar en el modelo. En el caso de escritura, en este módulo se asignan el nombre y la extensión que lleva el archivo que se genera en el sistema operativo. Los archivos de escritura y lectura se deben ubicar en el mismo directorio del modelo.

StateSet. En este módulo se crean grupos de estado de recursos que define el usuario, o se cambian los nombre de los estados predeterminados o automáticos (*Busy*, *Idle*, *Inactive* y *Failed*). Por ejemplo, si se quiere denominar a los estados automáticos (*Auto-states*) *Idle* y *Busy* como *Ocioso* y *Cortando*, esta opción se puede hacer en este módulo. En casos más avanzados pueden haber diferentes estados que define el usuario, por ejemplo, alistamiento, montaje y corte para un mismo recurso, dentro del estado automático *Busy*.

Statistic. Se encarga de especificar estadísticas definidas por el usuario acerca de entidades o de variables del modelo. Se pueden generar estadísticas sobre utilización de recursos, promedios ponderados en el tiempo de una variable, intervalos de tiempo para entidades, entre otros aspectos. Este módulo ofrece la opción de escribir las estadísticas recolectadas en un archivo con formato especial que se puede analizar mediante la herramienta *Output Analyzer*.

Storage. Define los sitios en los que se fija la animación de entidades o almacenamientos. Los almacenamientos se representan por una especie de fila, pero a diferencia de ésta, la representación de la entidad aparece sólo cuando se encuentra realizando diversas operaciones. En un almacenamiento se pueden tener estadísticas del número de entidades en un momento determinado.

3. VARIACIONES EN LA CAPACIDAD DE LOS RECURSOS

PROGRAMACIÓN (*SCHEDULE*)

ARENA[®] ofrece la posibilidad de establecer un patrón de cambios en la capacidad de los recursos.

Con base en el tiempo, los cambios de capacidad pueden tener una duración constante o aleatoria, y se repiten periódicamente en el modelo. También se ofrecen opciones de cómo se manejan los cambios en la capacidad una vez éstos ocurran (apropiativo, espera e ignora). Sólo se permite un horario por recurso; ejemplo, capacidad para procesar 5 unidades por 6 horas, 3 unidades por 2 horas, 1 unidad por 1 hora, etc.

FALLAS (*FAILURES*)

En algunos procesos industriales es necesario tener en cuenta las variaciones que se pueden presentar ante las fallas en las máquinas. En ARENA[®] se entiende como falla cualquier alteración que se presente en el normal funcionamiento de un recurso; por ejemplo, la vida útil de una determinada pieza en una maquinaria y la interrupción del servicio de un cajero en un banco.

Las fallas en ARENA[®] reducen la capacidad de un recurso a 0. Su manejo se hace mediante el módulo de datos *Failures*, en el cual se especifican los datos de probabilidad o tiempo de ocurrencia y duración de la interrupción.

Se inhabilita totalmente un recurso que se base en conteo de unidades o en tiempo. El manejo de la falla cumple las mismas reglas que los horarios (apropiativa, espera, ignora). Un recurso puede tener múltiples fallas; ejemplo, cambio de herramienta cada 100 piezas, 15 minutos de descanso del operario cada 2 horas, etc.

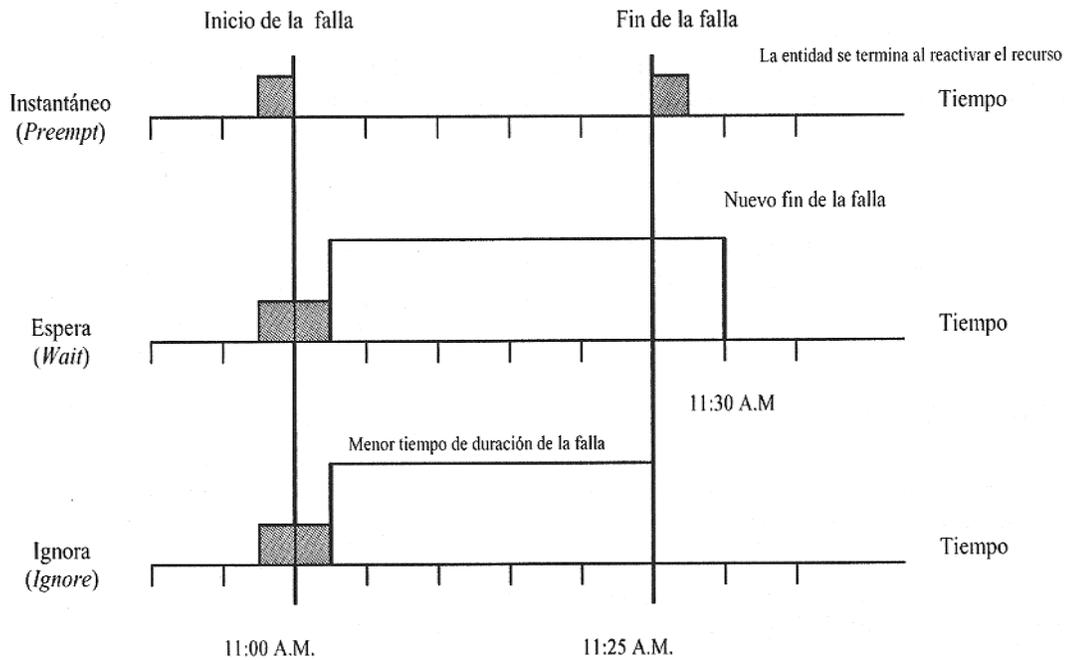
Manejo de fallas

Las fallas se pueden presentar durante el procesamiento de una pieza; cuando esto ocurre, ARENA[®] contempla varias posibilidades. A continuación se presenta una breve descripción de cada una de ellas:

Preempt (Apropiativa). La falla ocurre inmediatamente. Si se encuentra una entidad en proceso, ésta queda inconclusa y se termina cuando el recurso se reactive.

Wait (Espera). Si en el momento de ocurrir la falla una entidad se encuentra en proceso, ésta termina de procesar el recurso y entra en estado de inactividad. El tiempo de duración de la falla permanece constante.

Ignore (Ignora). La falla ocurre instantáneamente, y si se encuentra una entidad en proceso, ésta termina su operación, y este tiempo se desconecta de la duración de la falla. Esto hace que la duración de la falla, en algunas ocasiones, sea menor de lo que se especifica.



Ejemplo 6.1

■ Descripción

Fallas en los recursos. Reglas para controlar la capacidad

Las partes de un proceso entran con una distribución exponencial con media de 5 minutos; un operario las procesa en un turno en un tiempo normal con media de 4.5 minutos y desviación estándar de 0.95. Cuando se termina el proceso, las partes se retiran del sistema.

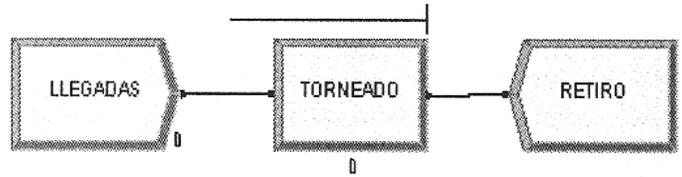
La vida útil del buril del torno es de 20 piezas; así que una vez se cumpla esta cuota de producción se debe detener la máquina y cambiar la pieza en una operación que demora un tiempo uniforme entre 6 y 10 minutos. Adicionalmente, el operario debe realizar limpieza a la máquina cada 2 horas. Cuando llega el momento de la limpieza, la máquina se encuentra procesando una pieza; de tal manera que suspende la operación y queda inactiva hasta cuando el operario realice la limpieza, la cual tiene una duración de 3 minutos. Pasado este tiempo, la máquina continúa su proceso.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create, Process, Dispose*
Módulos de datos: *Entity, Resource*
- Panel de procesos avanzados
Módulos de datos: *Failures*

■ Desarrollo

Se crea un modelo sencillo, así como se muestra en la siguiente figura, teniendo en cuenta la creación de un recurso llamado *torno* para efectuar la operación.



a. Declaración de fallas

En el módulo de datos del recurso se declaran las fallas que éste experimentará y la regla que gobierna el cambio de capacidad.



Resource

En la columna de fallas se agregan aquellas que experimentará el recurso; siempre se accede con un clic

Resource Basic Process								
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1	TORNO	Fixed Capacity	1	0.0	0.0	0.0		2 rows

Double-click here to add a new row.

Dentro del módulo de datos del recurso se encuentra el módulo de datos de la declaración de fallas de éste. Un recurso puede experimentar más de una falla. Para adicionar una falla se hace doble clic en la última línea; luego, del menú desplegable se escoge la regla que va a regir el cambio de capacidad del recurso, el cual, en el caso del ejemplo en estudio, obedece a *Preempt* para la falla que se basa en tiempo (limpieza). La otra falla, cambio de buril, debido a su naturaleza no requiere de ninguna regla específica, ya que el cambio de capacidad está claramente definido por el número de piezas.

Failures		
	Failure Name	Failure Rule
1	CAMBIO DE BURIL	Ignore
2	LIMPIEZA DE TORNO	Preempt

Double-click here to add a new row

- Wait
- Ignore
- Preempt

Regla escogida para el cambio de capacidad

El procedimiento anterior permite que las fallas se asignen a los recursos; las características de las fallas se ingresan al programa en la plantilla de procesos avanzados (*Advanced process panel*) del módulo de datos *Failure*.



Failure

	Name	Type	Up Time	Up Time Units	Count	Down Time	Down Time Units	Uptime in this State only
1	CAMBIO DE BURIL	Count	10	Hours	20	UNIF(6, 10)	Minutes	
2	LIMPIEZA DE TORNO	Time	2	Hours	3		Minutes	

Double-click here to add a new row.

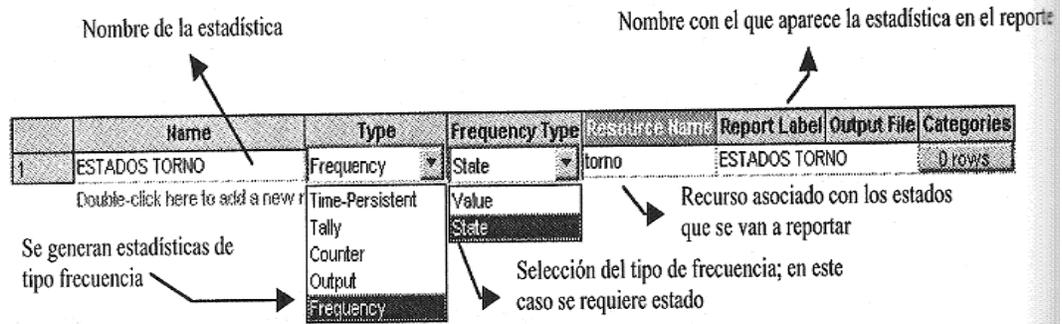
A continuación se explican cada una de las columnas que componen el módulo de datos *Failure*:

Campo	Descripción	Ejemplo
<i>Name</i>	Nombre de la falla. Este campo se puede descargar del menú desplegable, pues se encuentra ya definido en el módulo de datos del recurso correspondiente.	Cambio de buril, limpieza de máquina, mantenimiento.
<i>Type</i>	Tipo de falla. Aquí se especifica si la ocurrencia de la falla depende de un número determinado de piezas procesadas (<i>count</i>) o de una expresión de tiempo (<i>time</i>).	Cada 10 piezas (<i>count</i>) Cada 28 horas (<i>time</i>)
<i>Up time</i>	Tiempo entre fallas. Puede ser constante o puede tratarse de una distribución. Sólo se habilita cuando la falla se basa en tiempo.	Cada 28 horas (<i>time</i>). Tiempo entre fallas exponencial con media de 10 horas (<i>time</i>).
<i>Up time units</i>	Unidades del tiempo entre fallas	Horas, minutos, etc.
<i>Count</i>	Número de piezas que condicionan la aparición de la falla. Sólo se habilita cuando la falla se hace por conteo de entidades (<i>count</i>).	Cada 100 piezas
<i>Down time</i>	Duración de una falla que, una vez ocurrida, se basa en tiempo; puede ser una constante o una distribución.	Cada 28 horas (<i>time</i>) Expo (10) horas
<i>Downtime units</i>	Unidades de la demora de la falla	Horas, minutos, etc.
<i>Uptime this state only</i>	En este campo se define que un estado determinado del recurso cuente con tiempo hábil entre fallas. El tiempo transcurrido en otros estados no cuenta para la ocurrencia de la falla.	Falla cada 10 horas, las cuales transcurren en estado ocupado (<i>busy</i>).

Con este procedimiento se definen las fallas en los recursos, y al mismo tiempo se crea automáticamente un nuevo estado adicional a los que predetermina el programa, *Idle* (ocioso), *Busy* (ocupado). Este nuevo estado recibe el nombre de *Failed*, y es posible que se le asigne una figura, con el propósito de animar el paso del recurso a través de él de manera análoga a la animación de los estados ocupado y ocioso.

b. Estadísticas del estado de los recursos

Las estadísticas pertinentes al análisis del comportamiento de un recurso con respecto a diversos estados, como ocupación, ocio, fallas e inactividad (estados predeterminados) y algún otro que defina el usuario, se muestran en el reporte de frecuencias (*Frecuencias*), en el cual se presenta el porcentaje de tiempo que un recurso pasa en un estado específico. Para generar el reporte de frecuencias se debe recurrir al módulo de datos de estadísticas (*Statistics*) en el Panel de procesos avanzados.



Campo	Descripción
<i>Name</i>	Nombre de la estadística que se va a generar.
<i>Type</i>	Tipo de estadística que se va a generar según el propósito del usuario; en el caso del ejemplo se requiere la frecuencia.
<i>Frequency type</i>	Tipo de frecuencia. Puede estar relacionado con los estados de un recurso o con un valor.
<i>Resource name</i>	Nombre del recurso que está asociado con la estadística; sólo se activa cuando el tipo de frecuencia es <i>estado (state)</i> .
<i>Expression</i>	Expresión cuyo valor se evalúa; este campo sólo se activa cuando el tipo de frecuencia es <i>valor (value)</i> .
<i>Report label</i>	Nombre que recibe la estadística en el reporte; si no se especifica ninguno, el nombre (<i>name</i>) de la estadística es <i>usado</i> .
<i>Output file</i>	Este campo se usa cuando se desean guardar las estadísticas en un archivo válido del programa, con el fin de ser analizadas con el <i>Output Analyzer</i> . Si no se llena el campo, no se genera archivo alguno.
<i>Categories</i>	Hoja de trabajo adicional en la cual se definen las categorías o clases en las que se mide la frecuencia. No es necesario diligenciar esta hoja para el caso de los estados. En este campo se especifican los rangos que comprende cada categoría cuando se trata de frecuencias para valores.

En el caso de los estados, las estadísticas de frecuencias muestran el número de veces que un recurso visitó un estado, así como el tiempo promedio durante el cual permaneció en él; de igual manera, dan información sobre el porcentaje de tiempo en cada estado.

En lo que se refiere a los valores, se deben especificar categorías para cada rango o valor que se requiera analizar. Ejemplo, número de clientes en el sistema. Una categoría de 0-5 se puede denominar baja; de 6-10, media, y de 10-20, alta. Las estadísticas de frecuencias recolectarán información para cada categoría definida de la variable de manera análoga a los estados.

Adicionalmente, la base del cálculo del porcentaje de tiempo se puede modificar sin tener en cuenta (*Exclude*) una o más categorías (*Restricted percent*). Esto quiere decir que los porcentajes no se calcularán con base en el tiempo total simulado, sino con base en la diferencia entre el tiempo total y los tiempos transcurridos en cada categoría excluida.

c. Reportes

Cuando la simulación se corre por un tiempo de 72 horas, se genera el siguiente reporte:

- Category Overview
- Category by Replication
- Entities
- Frecuencias**
- Processes
- Queues
- Resources
- Transfers
- User Specified

En los reportes se selecciona el estado que corresponde a las frecuencias (*Frecuencias*)

Nombre de la estadística	Número de veces que visitó el estado correspondiente	Tiempo medio en el estado correspondiente	Porcentaje del tiempo total en el estado correspondiente	
			Standard Percent	Restricted Percent
ESTADOS TORNO				
BUSY	15	0.07561017	1.58	1,58
FAILED	} Estados que experimentó el recurso	35	0.05000000	2.43
IDLE				

Cuando un estado definido no ocurre o tiene una duración de 0, éste no se muestra en el reporte

Cuando no se excluye ninguna categoría, el porcentaje restringido es igual al estándar

4. EMPLEO DE ENTIDADES DE CONTROL

4.1. CÓMO RETENER ENTIDADES Y LIBERARLAS CON UNA SEÑAL

Existen situaciones en las cuales el flujo normal de un producto (entidad) es interrumpido hasta cuando se produce un determinado hecho. Esta interrupción, por lo general, exige que el producto se detenga en un espacio físico (cola), lo cual provoca una demora, que, en algunos casos, forma parte del proceso normal.

ARENA[®] permite manejar este tipo de situaciones a través de la implementación del módulo lógico *Hold*, el cual cumple tres funciones:

- Esperar una señal (*Wait for signal*)
- Examinar una condición (*Scan for condition*)
- Esperar de manera infinita (*Infinite hold*)

En esta sección se estudia la primera función; las dos últimas se explican más adelante.

La función *Wait for signal* del módulo *Hold* se encarga de retener entidades que son liberadas con la implementación obligatoria del módulo *Signal*, el cual emite una señal para la liberación. Las entidades entran al módulo *Hold*, que está configurado con esta función, y se retienen hasta que la señal de liberación se envíe desde otra instancia del modelo.

Campo	Descripción	Ejemplo
<i>Name</i>	Nombre del módulo; debe ser único.	Zona de espera
<i>Type</i>	Función del módulo; puede ser de tres tipos: esperar una señal, examinar una condición o esperar de manera infinita.	Espera por señal
<i>Wait for value</i>	Identificación de la señal que se espera. Este campo resulta útil para diferenciar las señales cuando se emplean más de una función <i>Esperar una señal</i> (sólo se activa cuando el tipo es <i>Wait for signal</i>). Debe ser numérico.	1
<i>Limit</i>	Cantidad máxima de entidades que luego de encontrarse retenidas se liberan al recibir la señal.	5
<i>Queue type</i>	Tipo de cola; puede ser interna (no se muestran y no se recolectan estadísticas), normal (<i>Queue</i>), grupo de colas (<i>Set</i>), indexada a una cola específica con base en un atributo o en una expresión (<i>Attribute / Expression</i>).	<i>Queue</i> (cola normal)
<i>Nombre de la cola</i>	Nombre de la cola (es activo si el tipo de cola es <i>Queue</i>). Por lo general, el nombre de la cola se genera automáticamente con el nombre del módulo, sin embargo, el usuario puede cambiarlo.	Zona de espera.Queue

El módulo *Signal* envía la señal de liberación al módulo o módulos que contienen entidades que estén esperando. Para esto se requiere el paso de entidades a través de este módulo; las entidades que manejan el flujo de otras en el sistema se llaman *entidades de control*. La señal enviada la reciben solamente los módulos *Hold* que contengan el mismo valor de señal que se especifica en el campo *Signal value*. La cantidad máxima de entidades que se va a liberar será el menor de los límites declarados en ambos módulos; esto si las entidades en espera alcanzan estas cantidades; de otra forma, se liberan las que estén disponibles.

Campo	Descripción	Ejemplo
<i>Name</i>	Nombre del módulo; debe ser único.	Señal (el programa no acepta ñ)
<i>Signal value</i>	Valor de la señal. La señal se enviará a los módulos <i>Hold</i> (configurados como <i>Wait for signal</i>) que tengan este mismo valor en el campo <i>Wait for signal</i> . La señal se enviará a todos los módulos <i>Hold</i> de este tipo, incluso a los que se encuentren en submodelos.	1 (se enviará a todos los módulos <i>Hold</i> con valor 1 en <i>Wait for signal</i>).
<i>Limit</i>	Valor máximo de entidades que se van a liberar.	10

En caso de que se tenga más de un módulo *Hold* que esté esperando por la misma señal, la cantidad que se va a liberar, especificada en el módulo *Signal*, se distribuye entre todos los módulos *Hold* relacionados, teniendo en cuenta el número máximo que se va a liberar en cada uno y empezando por el primero de éstos. Con el fin de ilustrar mejor el funcionamiento de los módulos *Wait for signal*, a continuación se exponen dos posibles situaciones.

- *Situación 1.* Un módulo *Signal* tiene valor de señal 1 y cantidad que se va a liberar de 10. Se encuentran tres módulos *Hold*, cada uno con 20 unidades retenidas; valor de señal 1 y límite de 5 unidades.

Resultado: El módulo *Signal*, de 20 entidades, libera 5 del primer módulo; de 20 entidades, 5 del segundo módulo, y ninguna del tercero.

- *Situación 2.* Un módulo *Signal* tiene valor de señal 1 y cantidad que se va a liberar de 12. Se encuentran tres módulos *Hold*, cada uno con 20 unidades retenidas; cada uno tiene valor de señal 1 y límites de 5, 6 y 3 unidades, respectivamente.

Resultado: El módulo *Signal* libera, de 20 entidades, 5 del primero; de 20 entidades, 6 del segundo (van 11 entidades liberadas de 12), y sólo 1 del tercero (para llegar a 12), aunque pueden ser más (hasta 3).

Ejemplo 6.2

■ Descripción

Uso de entidades de control. Cómo retener entidades y liberarlas con una señal

Con base en el ejemplo anterior, asuma que las partes entran al sistema y esperan 2 horas para que sean liberadas en grupos de 10 y permitan su entrada al proceso. Una vez se liberan, el operario las procesa en el turno.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create, Process, Dispose*
Módulos de datos: *Entity, Resource*
- Panel de procesos avanzados
Módulos lógicos: *Hold, Signal*

■ Desarrollo

Con base en el ejemplo 6.1, se deben retener las entidades entrantes justo cuando salen del módulo *Create*. Para esto se agrega un módulo *Hold* de tipo *Wait for signal*, y se le da el nombre de *Espera*.

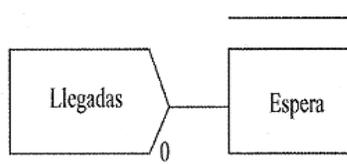
Tipo de módulo *Hold*. En este caso la entidad espera por una señal (*Wait for signal*)

Máximo de entidades que se van a liberar

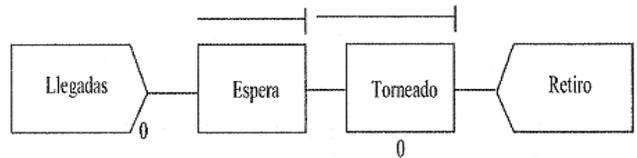
Valor de la señal. Este campo-valor identifica la señal y la asocia con el módulo *Signal*

Tipo de cola en la cual las entidades van a esperar por la señal

Después que se agrega el módulo *Hold*, el sistema se observa de la siguiente manera:



Posteriormente, y así como se señala en la sección anterior, se sigue con el proceso en la implementación de los módulos *Process* y *Dispose*.



Las entidades entrantes se quedan en la cola del módulo *Wait* hasta que reciban la señal; por lo tanto, se necesita de lógica externa para enviar la señal de liberación. Con este fin se hace una entrada adicional que genere una entidad cada 2 horas y envíe una señal que libere 10 entidades.



Create

Entrada de tipo constante cada 2 horas

Tiempo de la primera creación. Las unidades son las mismas que se especifican en los datos de tiempo entre llegadas

Se procede, entonces, a abrir el módulo *Signal*, en el cual se especifica la cantidad de entidades que se van a liberar:



Signal

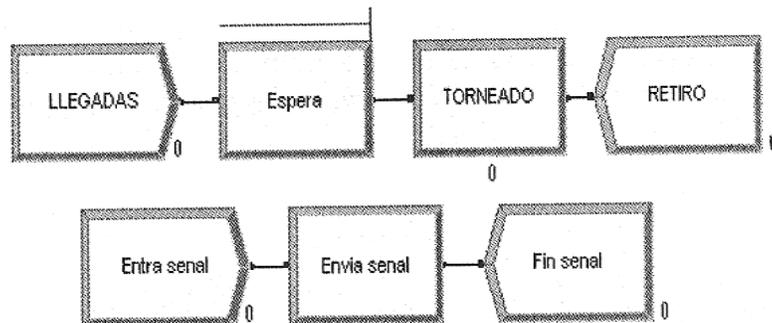
Valor de la señal, vínculo o asociación con un módulo *Wait*

Número de entidades que se va a liberar en el (los) módulo(s) *Wait* con el mismo valor de señal

La hoja de trabajo que corresponde a este módulo se muestra a continuación:

Signal - Advanced Process			
	Name	Signal Value	Limit
1	Envía señal	1	10

Una vez se ha introducido y editado el módulo *Signal*, se abre un módulo *Dispose*, el cual especifica el fin de la señal. El sistema terminado se observa así:



4.2 CÓMO RETENER ENTIDADES Y LIBERARLAS CUANDO UNA CONDICIÓN ESPECÍFICA SE CUMPLA

Retener entidades en un sitio determinado y liberarlas sólo cuando una condición específica se cumpla es una situación bastante común dentro un sistema. A continuación se presentan algunos ejemplos:

- En un modelo de manufactura se dejan pasar piezas mientras haya capacidad de atención en una máquina.

- Se retiene un pedido, en un sistema de inventario, hasta que se alcance el punto de reorden.
- En un sistema de inspección continuo se espera una secuencia de 10 unidades conformes para cambiar el tipo de muestreo.

Cuando se presenta cualquiera de estos casos es necesario recurrir a la función *Scan for condition* del módulo *Hold*, el cual examina todo el sistema y retiene la entidad hasta cuando se cumpla una condición determinada para liberarla. En el caso de que varias entidades se encuentren esperando una señal, la primera de ellas se libera y recorre los módulos lógicos posteriores hasta cuando sea detenida. Este procedimiento se puede dar en una demora (recurso con un tiempo de proceso) u otro módulo *Hold* (módulos como *Decide*, *Assign*, entre otros, no detienen la entidad); entonces, el programa vuelve a evaluar la condición para la próxima entidad que espera en el módulo, la cual se puede liberar o no de acuerdo con el valor de la condición, a diferencia de la acción *Wait*, que libera más de una entidad al mismo tiempo cuando ocurre un evento de señal.

Ejemplo 6.3

Uso de variables predeterminadas (expresiones). Cómo retener entidades y liberarlas cuando una condición específica se cumpla

■ Descripción

Determinadas partes entran con una distribución exponencial de media de 5 minutos, y sólo se liberan cuando el recurso *torno* tiene capacidad menor de 3. El recurso *torno* procesa unidades en un tiempo normal con media de 12 minutos y desviación estándar de 0.95; su máxima capacidad es 3. Después que las entidades se procesan, éstas se retiran del sistema.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create*, *Process*, *Dispose*
Módulos de datos: *Entity*, *Resource*, *Schedule*
- Panel de procesos avanzados
Módulos lógicos: *Hold*

■ Desarrollo

Una vez el módulo *Create* se ha editado de acuerdo con las especificaciones del enunciado, se procede a abrir el módulo *Hold*:



Hold

Tipo de módulo. En este caso se desea retener una entidad y liberarla sólo bajo ciertas condiciones

Condición que se va a evaluar. Cuando la condición es verdadera, la primera entidad del módulo se libera. La condición se evalúa nuevamente cuando la entidad liberada se detiene

En el módulo anterior se utiliza una expresión compuesta por variables que predetermina el programa: $NR(\text{torno}) < MR(\text{torno})$.

*NR (nombre del recurso)*⁴: Esta variable devuelve el número de unidades de capacidad ocupadas del recurso especificado. El argumento es el nombre del recurso que se desea examinar.

Expression builder: *Basic process variables/Resource/Usage/Current number busy*

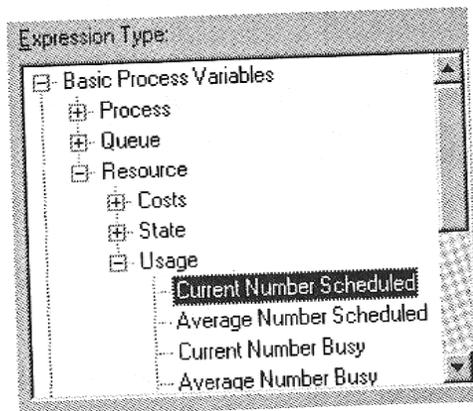
⁴ Estas expresiones se pueden construir mediante el asistente que se utiliza con este propósito (*Expression builder*), haciendo clic derecho en el campo *Condition* (ver capítulo 4, *Animación de variables*).



Ejemplo. Si en el caso en estudio el turno tiene capacidad de 3, el valor de la variable $NR(\text{turno})$ varía entre 0 y 3 de acuerdo con su ocupación; si la variable está ociosa, asume el valor 0 (ninguna unidad ocupada).

MR (*nombre del recurso*)⁵: Esta variable devuelve un número que corresponde al máximo de unidades de recurso disponibles. Cabe anotar que éste se especifica en tiempo de diseño en el campo *Capacity* del módulo de datos *Resource*; también puede definirse mediante *Schedule*. El argumento es el nombre del recurso que se desea examinar. El usuario puede cambiar el valor de esta variable durante la simulación (en tiempo de ejecución); un valor de 0 para esta variable deja al recurso inactivo.

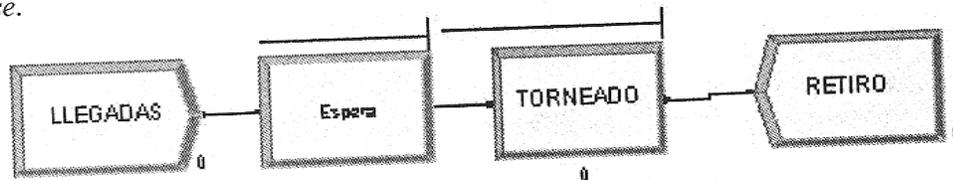
Expression builder: *Basic process variables/Resource/Usage/Current number schedule*



Ejemplo. En el caso en estudio, la variable MR (*turno*) asume el valor de 3 durante la simulación, ya que no se cambia durante toda la corrida.

La expresión construida permite que una entidad se libere sólo si hay capacidad disponible en el recurso *turno*. Esta situación se mantiene, incluso, si llega a cambiar el número máximo de unidades de recurso disponible.

Cuando culmina la edición del módulo *Hold*, se termina el ejercicio mediante el procesamiento de la entidad en el módulo *Process* y su retiro del sistema en el módulo *Dispose*.



⁵ Estas expresiones se pueden construir mediante el asistente que se utiliza con este propósito (*Expression builder*), haciendo clic derecho en el campo *Condition* (ver capítulo 4, *Animación de variables*).

5. CÓMO BUSCAR Y REMOVER ENTIDADES ESPECÍFICAS DE UNA COLA

Tanto en el medio industrial como en el de servicios es común encontrarse con situaciones en las cuales es necesario que, de un determinado grupo o cola, se realice una selección de entidades con características específicas. Por ejemplo, a la dirección de una planta procesadora de café le interesa separar los granos más grandes para exportarlos; éstos se separan de los demás para seguir un proceso de refinamiento. Esta selección se realiza mediante la implementación de tres módulos, a saber:

- a. *Hold*. Retiene las entidades (productos o clientes).
- b. *Search*. Busca el conjunto de entidades que cumplan con las condiciones especificadas. Este módulo maneja tres opciones:
 - *Search a queue*
 - *Search a batch*
 - *Search an expresión*

El módulo *Search* retorna, dentro de la variable *Global J*, la posición en la cola de la primera entidad que cumple con las condiciones especificadas. En esta sección sólo se explica la primera opción, *Search a queue*.

- c. *Remove*. Saca de la cola a las entidades seleccionadas y las envía al recurso que las va a procesar.

Ejemplo 6.4

Cómo buscar y remover entidades específicas de una cola

■ Descripción

A una zona de espera entran 50 partes con una distribución exponencial con media de 5 minutos. Las piezas tienen un diámetro que sigue una distribución uniforme entre 10 y 20 cm. De estas piezas, sólo se seleccionan aquellas que tienen un diámetro menor a 15 cm; el resto permanece en la bodega. Las piezas seleccionadas se procesan en el recurso *torno*, el cual tiene una capacidad fija de 3 y procesa unidades en el tiempo normal con media de 12 minutos y desviación estándar de 0.95. Cuando salen del torno, estas piezas se retiran del sistema.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create, Process, Dispose*
Módulos de datos: *Entity, Resource*
- Panel de procesos avanzados
Módulos lógicos: *Hold, Remove*

■ Desarrollo

De una sola vez se generan 50 llegadas mediante un módulo *Create*, así como se muestra a continuación:



Create

Se generan llegadas de tamaño 50

Se restringe el número de llegadas a 1

Se especifica que la primera llegada ocurra en el tiempo 0

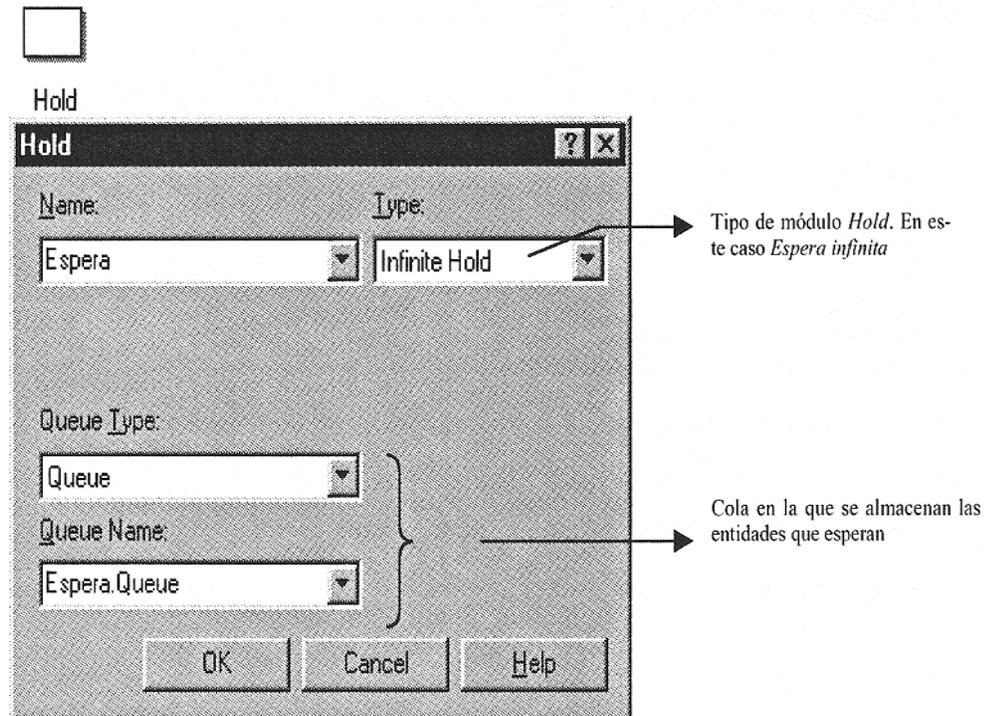
Las entidades entrantes deben tener un atributo llamado *diámetro*, cuyo valor sigue una distribución uniforme de acuerdo con lo que se especifica en el enunciado. Este atributo y su respectivo valor se crean en un módulo *Assign*:



Assign

Se crea un atributo llamado *diámetro*, cuyo valor sigue una distribución igual a la especificada

Según las condiciones que se estipulan en el ejemplo, las entidades ya creadas se dirigen a un almacenamiento indefinido para esperar a que sean removidas. Esta espera se modela mediante un módulo *Hold* de tipo espera infinita (*Infinite hold*). Esta función del módulo mencionado almacena las entidades en una cola de manera indefinida en el tiempo; se hace necesario que para liberarlas se emplee una secuencia lógica externa.



A continuación se muestra el diagrama del modelo en ARENA®:



Empleo de la entidad de control

En casos como el que se está estudiando, en los que se tiene una espera indefinida, se hace necesario emplear una lógica independiente para que tome el control del sistema. En este ejemplo se debe crear una entidad que inspeccione cada posición de la cola y revise que la condición especificada se cumpla. En caso de que esta condición se cumpla, se debe remover la entidad de esa cola y retirarla del sistema.

Creación de la entidad de control

Este procedimiento se realiza mediante un módulo *Create* que genere una entidad en el tiempo 0 y con un máximo de llegadas de 1, así como se presenta a continuación:



Create

Creación de una entidad por llegada

Generación de máximo una llegada

Tiempo de la primera llegada

La entidad de control debe esperar a que haya una o más entidades en cola para evaluar y retirar alguna.

Importante. Si se intenta remover una entidad de una cola vacía, se ocasionará un error en tiempo de corrida. Es recomendable que, antes de realizar esta operación, se verifique si en la cola hay elementos.

Para verificar que la cola no esté vacía, se emplea un módulo *Hold* que esté configurado como *Scan for condition*, en el que sólo se permite el paso de la entidad si hay más de una en la cola *Espera.Queue*

Se emplea un módulo *Scan* para permitir el paso de entidades sólo si hay entidades en cola

Verifica que el número en la cola de espera sea mayor que 0 (es decir que haya algo en cola)

Una vez se ha verificado la condición de la cola, se procede a realizar la búsqueda de entidades específicas mediante el módulo *Search*.



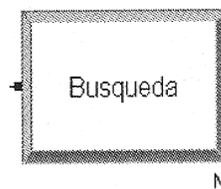
Search

Módulo *Search* de búsqueda en una cola

Nombre de la cola en la que se efectúa la búsqueda

Posición de inicio y fin de la búsqueda en la cola. Esta expresión se puede desarrollar a través del uso del asistente para la construcción de expresiones (*Expression builder*), el cual se referencia en el capítulo 4

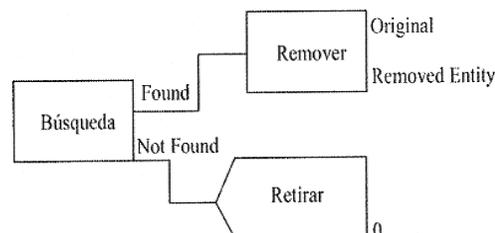
Condición que se evalúa



Camino que sigue la entidad de control en caso de que encuentre una entidad que cumpla con la condición

Camino que sigue la entidad de control en caso de que ésta no encuentre entidades que cumplan con la condición,

Con base en el ejemplo en estudio, se pedirá que la entidad de control remueva de la cola las entidades que satisfagan la condición especificada en caso de encontrar este tipo de entidades. Si no se encuentra ninguna entidad que cumpla con la condición en la cola, la entidad de control se retirará del sistema y terminará la simulación.

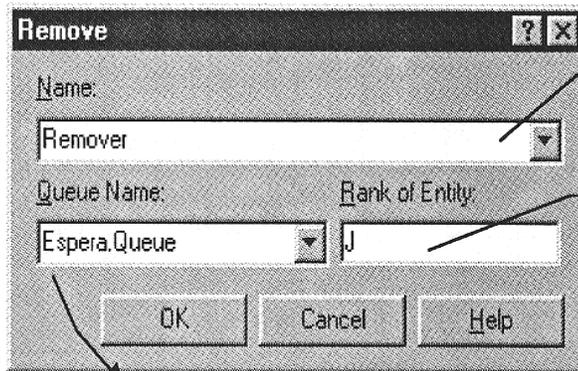


La entidad se retira mediante un módulo *Dispose*.

A continuación se muestra la edición del módulo *Remove*, el cual se encarga de retirar de la cola la entidad que cumple con la condición:



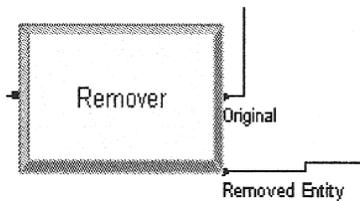
Remove



Nombre del módulo

Posición en la cual se encuentra la entidad que se desea remover. Se coloca *J*, por que, como se señaló anteriormente, en esta variable se almacena la posición de la primera entidad que cumple con la condición especificada

Nombre de la cola de la cual se retiran las entidades

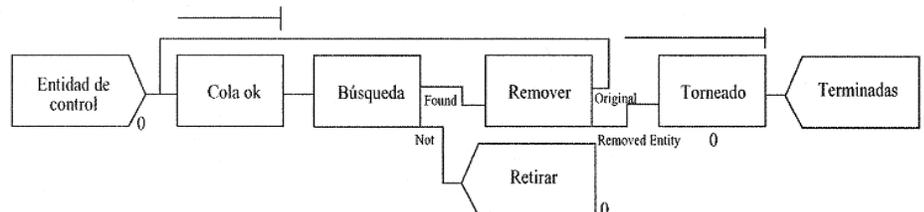


Camino que sigue la entidad que remueve o entidad de control. Esta tiene prioridad sobre la removida (saldrá primero del módulo); continuará hacia los módulos posteriores hasta que sea detenida

Camino que sigue la entidad removida después de que la original ha sido detenida en su paso

La entidad original se conecta al módulo *Scan* del principio para verificar nuevamente si hay cola. Esto permite que, mientras haya cola, la entidad de control siga buscando hasta que no encuentre ninguna entidad con la condición especificada.

A continuación se muestra el diagrama del modelo en ARENA®:

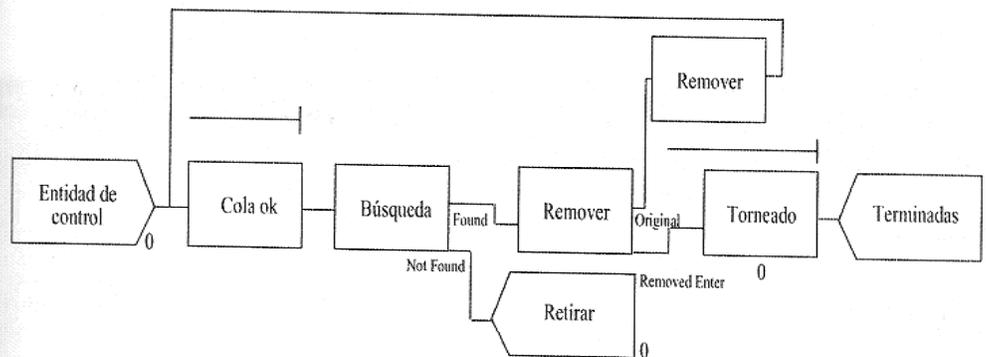


Bajo el esquema anteriormente descrito, la entidad de control nunca encuentra una demora hasta que termina la búsqueda en la cola; esto ocurre cuando no queda ninguna entidad que cumpla la condición enunciada. Las entidades removidas esperan en una

cola interna que se encuentra en la salida del módulo *Remove*, y se liberan cuando la entidad de control pierde el enfoque, es decir, es demorada, retenida o, en este caso, retirada del sistema.

Si se desea ver el efecto que produciría una demora en el camino de la entidad de control (pérdida del enfoque), se puede introducir una instrucción *Delay* (demora) en la salida del módulo *Remove* para detener la entidad de control por un momento; la demora puede ser de 1 minuto, por ejemplo. De esta manera, cada entidad removida de la cola se deposita inmediatamente en el módulo *Process* (ya que la demora hace que la entidad de control pierda el enfoque) y traslada la prioridad a la entidad removida.

A continuación se muestra el diagrama de la situación expuesta:



6. MODELO DE ENSAMBLE

En el medio real existen diferentes operaciones en las que se une un determinado número de componentes para fabricar un producto final; esto corresponde a una operación de ensamble. En ARENA® los módulos *Match* y *Batch* forman la estructura lógica básica para modelar la unión de piezas.

La función del módulo *Match* es sincronizar el número de entidades que se le indique. Esto quiere decir que si este número es tres (3), por ejemplo, el módulo genera tres colas (una para cada componente), y sólo libera la primera entidad de cada cola cuando las tres estén disponibles. Cabe anotar que la condición de un ensamble implica que de las tres se debe formar un componente.

Si la regla de sincronización es por atributo, entonces, las entidades se acumulan en cada fila, y sólo serán liberadas cuando en cada fila haya una con el mismo valor de atributo, sin importar su posición dentro de ésta.

Pueden haber entidades con diferentes valores de atributos en la misma cola; sin embargo, el módulo *Match* hace coincidir las que tengan el mismo valor para el atributo especificado.

Módulo Match

Campo	Descripción
Name	Nombre del módulo
Number to match	Número de entidades que se va a sincronizar; se genera una cola para cada entidad.
Type	Tipo de regla de sincronización; puede ser para cualquier entidad (<i>any entity</i>) o por atributos (<i>based on attribute</i>).
Attribute name	Nombre del atributo que se utilizará como referencia para sincronizar las entidades entrantes. Sólo se activa cuando el tipo de regla se basa en atributos (<i>based on attribute</i>).

El módulo *Batch* forma un lote o grupo de tamaño permanente, el cual está representado por una entidad (componente ensamblado) con el número de entidades que lleguen a él. El número de entidades que se va a agrupar debe concordar con el número de entidades que se va a sincronizar, el cual se especifica en el módulo *Match*. Para resumir este procedimiento se puede señalar que el módulo *Match* se encarga de ensamblar las entidades de acuerdo con el número especificado y de liberarlas hacia el módulo *Batch*, el cual las une en un grupo permanente formado por una sola entidad.

Ejemplo 6.6

■ Descripción

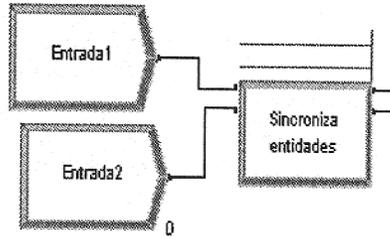
En un proceso de ensamble hay dos entradas: una para las partes de tipo A, la cual sigue una distribución exponencial con media de 20 minutos, y otra para las partes de tipo B, que tiene un tiempo entre llegadas constante de 30 minutos. Considere que una parte de tipo A se une con una de tipo B para que puedan ser retiradas del sistema.

■ Desarrollo

Se generan las llegadas mediante dos módulos *Create* separados, así como se muestra a continuación:



Posteriormente se adiciona un módulo *Match*, el cual se encarga de sincronizar las entidades que procedan de cada llegada. Este módulo acumula las entidades de más llegadas y espera hasta que llegue una de otro tipo para liberar los dos tipos a la vez.



A continuación se muestra la edición del módulo *Match*:

Número de entidades que se va a sincronizar

Tipo de módulo *Match* que se desea. No evalúa las entidades que le llegan, pero puede hacerlo en el caso de los atributos

De esta manera se garantiza que una entidad de cada tipo llegue al módulo *Batch*. A continuación se adiciona el módulo *Batch*:

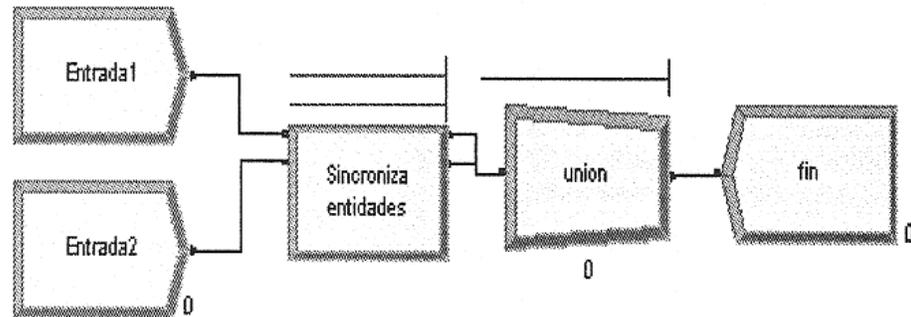


Se forma un grupo de tamaño 2 según la cantidad sincronizada en el módulo *Match*

Para mayor detalle acerca de la edición de este módulo, vea la sección *Formación y desarme de un lote de transferencia* en el capítulo 2

Finalmente se adiciona un módulo *Dispose*; este procedimiento permite que las entidades ensambladas sean retiradas del sistema.

A continuación se muestra el diagrama del modelo:



7. CÓMO GUARDAR ESTADÍSTICAS EN UN ARCHIVO EXTERNO

Muchas veces se hace necesario recolectar estadísticas sobre determinada variable con el fin de realizar pruebas estadísticas en otros programas. Entonces se requiere guardar estadísticas en archivos externos para que posteriormente se puedan procesar. En ARENA[®] es posible tanto leer archivos como escribir o generarlos mediante el módulo *Read/Write*.

El programa ofrece la posibilidad de generar archivos en hoja de cálculo (WKS); sin embargo, tiene sus limitaciones por la cantidad de observaciones que soporta.

En general, en un archivo se pueden escribir variables o atributos; el número de observaciones dentro del archivo corresponde al número de entidades que pasa por el módulo en cuestión.

Ejemplo 6.7

■ Descripción

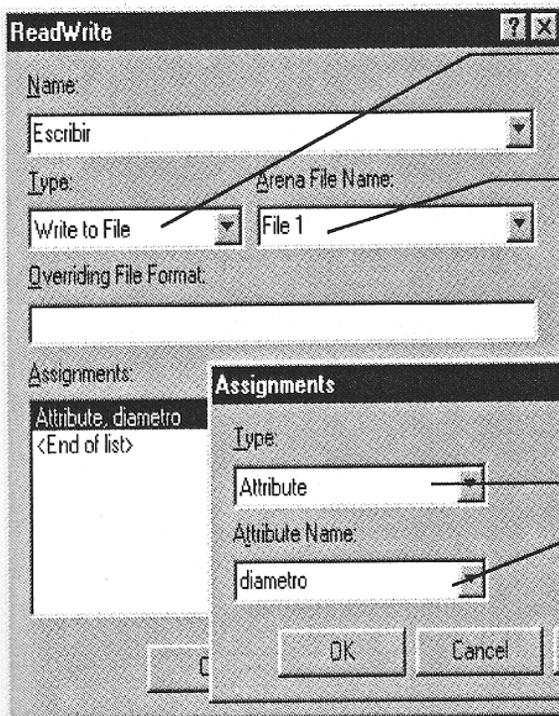
Considere que en el caso del ejemplo 6.4 se desea guardar, en un archivo externo, el atributo *peso* de las entidades que son mayores de 15.

■ Desarrollo

Inicialmente se adiciona un módulo *Read/Write* en la salida del módulo *Remove*, con el fin de contar las entidades que cumplan con la condición especificada.



ReadWrite



Se escoge el tipo del módulo. En este caso la opción es escribir a un archivo (*write to file*)

Arena® File Name es el nombre que se utiliza cuando se va a hacer referencia al mismo *Archdhised*

Se escoge el atributo que se desea escribir en el archivo externo

Adicionalmente, se debe abrir el módulo de datos *File* para dar nombre y extensión al archivo.



File

	Name	Operating System File Name	Structure	End of File Action	Comment Character
1	File 1	Pesos.dat	Free Format	Dispose	No

Double-click here to add a new row.

Name. Nombre que el programa da al archivo; es el mismo que se coloca en el campo *Arena® File Name* del módulo *Read/Write*.

Operating System File Name. Nombre y extensión con que aparece el archivo en *WINDOWS*. El archivo se genera en el mismo directorio en el que se encuentra el modelo, y se puede abrir con los programas *Notepad* o *Wordpad*.

Structure. Se recomienda el formato libre (*Free format*), ya que éste es el que permite mayor capacidad de escritura sobre un archivo.

8. USO DE ALMACENAMIENTOS

En ARENA[®] los almacenamientos tienen una aplicación diferente a la que usualmente remite el significado de este concepto. Se usan cuando una entidad ingresa a una serie de módulos lógicos que llevan un tiempo asociado, pero si se desea, el usuario puede ver la animación de la entidad fija en un sitio específico mientras estas instrucciones se ejecutan. Se trata más de un recurso de animación que de un almacenamiento físico como tal; sin embargo, es posible que para efectos de control de la lógica se vea la representación de la entidad en un lugar fijo mientras se le practican distintas operaciones. Adicionalmente, y según la recursividad del usuario, de la duración de la entidad.

Una clara utilidad de los almacenamientos se evidencia cuando se usan los módulos *Delay*, ya que en éstos la entidad experimenta una demora pero no puede ser visualizada mientras ocurre.

Módulo Store

Campo	Descripción
<i>Name</i>	Nombre del módulo
<i>Type</i>	Tipo de almacenamiento. Puede ser convencional (<i>Storage</i>), por medio de grupos, a través del uso de un atributo como índice (<i>Attribute</i>) o de una expresión como índice (<i>Expresión</i>).
<i>Storage name</i>	Nombre del almacenamiento. Sólo se activa cuando el tipo es <i>Storage</i> .

Módulo Unstore

Campo	Descripción
<i>Name</i>	Nombre del módulo
<i>Type</i>	Tipo de almacenamiento del cual se retira la entidad de acuerdo con la clase de almacenamiento que se le dio.
<i>Storage name</i>	Nombre del almacenamiento del cual se va a retirar la entidad. Sólo se activa cuando el tipo es <i>Storage</i> .

Ejemplo 6.8

■ Descripción

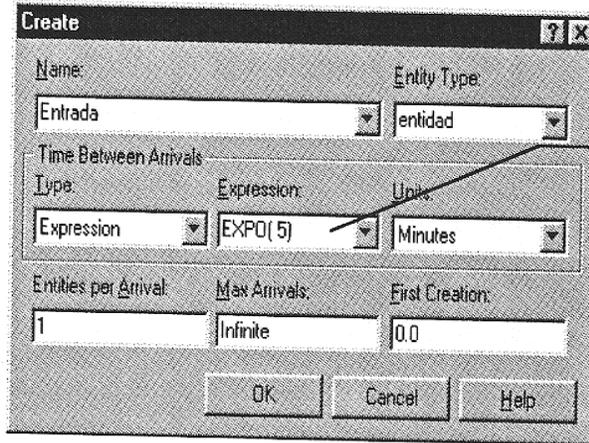
Considere una entrada exponencial con media de 5 minutos; las entidades entrantes experimentan una demora constante de 12 minutos, y posteriormente salen del sistema.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create, Process, Dispose*
Módulos de datos: *Entity, Resource*
- Panel de procesos avanzados
Módulos lógicos: *Store, Unstore*
Módulos de datos: *Storage*

Desarrollo

Primero se generan la llegadas mediante un módulo *Create*, así como se muestra a continuación.

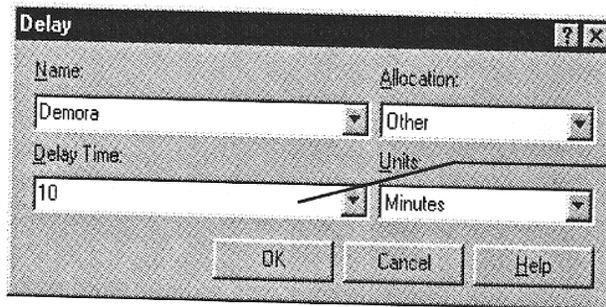


Se genera la llegada, así como se indica en el enunciado

Luego se adiciona un módulo *Delay*; en éste la entidad experimenta la demora que se especifica en el enunciado.

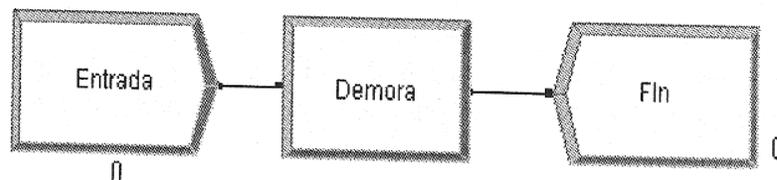


Delay



Se inserta un módulo *Delay* de acuerdo con las condiciones especificadas

Al final del modelo se coloca un módulo *Dispose* con el fin de retirar las entidades del modelo. A continuación se muestra el diagrama con los módulos mínimos requeridos:



De esta manera, cuando una entidad entra en el módulo *Delay*, experimenta una demora pero su animación no se ve. Para visualizar la animación o figura de la entidad que ingresa a este módulo se emplean los módulos *Store* y *Unstore*.

El módulo *Store* se coloca antes de que la entidad ingrese a la instrucción *Delay*, para indicar que la figura de esta entidad se va a visualizar en un sitio fijo (almacenamiento) a partir de ese punto del modelo.



Store

Tipo de módulo *Store*. En este caso se trata de un almacenamiento común (*Storage*)

Nombre del almacenamiento. Permite identificar el sitio en donde se localizan las entidades

Después que termina la demora en el módulo *Delay*, la animación de la entidad se libera con un módulo *Unstore*:

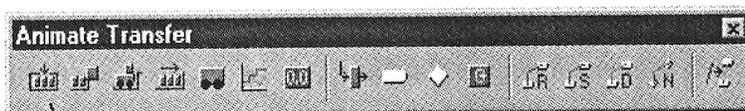


Unstore

Tipo de módulo *Unstore*. En este caso se libera un almacenamiento simple

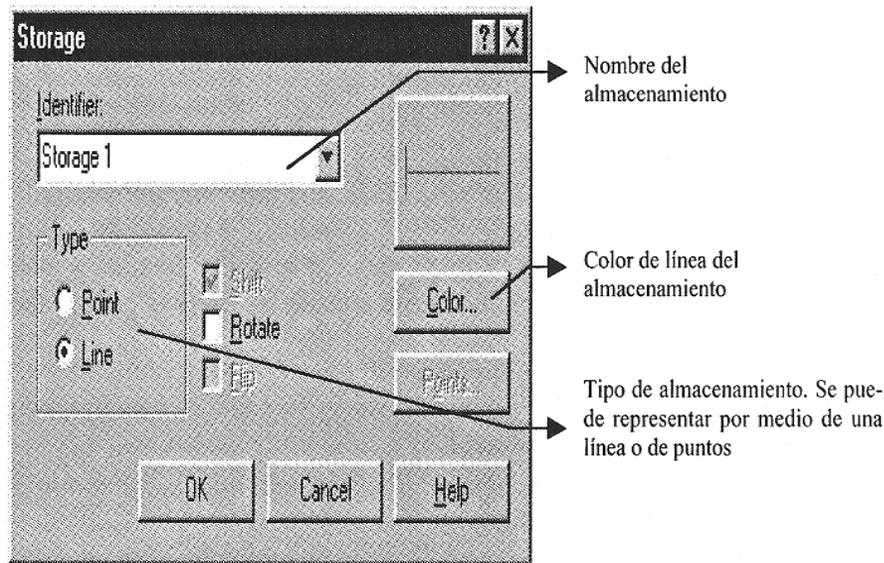
Nombre del almacenamiento del cual se libera la animación de la entidad

El sitio en el cual se desea visualizar la animación de las entidades, el almacenamiento, se debe crear por medio del botón *Storage* de la barra *Animate Transfer*:



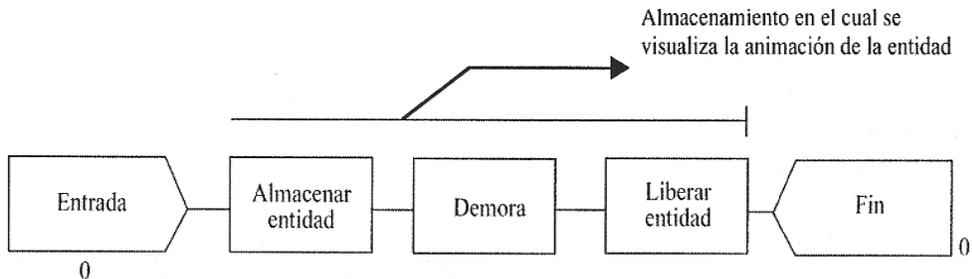
Almacenamiento

Cuando se hace clic en el botón de almacenamiento, aparece el siguiente formulario:



Cuando se hace clic en el botón *ok*, el puntero del mouse se transforma en cruz con el fin de que se ubique el almacenamiento en el sitio deseado. Este procedimiento se hace con clic sostenido si es un almacenamiento de tipo línea; en caso de que sea uno de tipo punto, se hace un clic por cada posición; para finalizar se hace doble clic.

A continuación se muestra el diagrama del modelo:



9. CÓMO RECOGER UN GRUPO DE ENTIDADES EN UNA COLA Y DEPOSITARLO EN OTRA PARTE DEL MODELO

En la construcción de modelos se pueden presentar situaciones en las cuales es necesario seleccionar un grupo de entidades de un conjunto de mayor tamaño para trasladarlo a otras instancias del modelo ARENA®. Esta aplicación es posible a través de la implementación ordenada de los módulos *Hold*, *Search*, *Pickup* y *Dropoff*. Ejemplos típicos de esta situación se enuncian a continuación:

- Selección de una muestra de tamaño 5 para hallar la media y la desviación estándar.
- En un sistema de transporte, cuando una ruta de bus específica llega a una parada, sólo los pasajeros que se encuentren esperando el transporte de esa ruta ingresan a los respectivos vehículos.

Los módulos *Pickup* y *Dropoff* se pueden emplear para remover entidades de una cola, con ciertas diferencias con respecto al módulo *Remove*, ya que este último sólo retira entidades de una cola para trasladarlas individualmente a módulos subsiguientes.

El módulo *Pickup* permite que una entidad de control recoja, de una cola, una o más entidades en un rango determinado; por ejemplo, de la cola recoge entidades de la 1 hasta la 5. Este módulo forma un grupo con las entidades recogidas, el cual está representado por aquella que recoge (entidad de control). La dimensión o tamaño del grupo se almacena en la variable NG (*Number in Group*). Una vez se ha formado este grupo, las entidades recogidas se remitirán implícitamente a donde se envía la entidad de control.

Con el fin de descargar una o varias entidades del grupo formado, se emplea la instrucción *Dropoff*. En este módulo se especifica la cantidad de entidades que se va a descargar y la posición dentro del grupo; el grupo se puede considerar como una sucesión de entidades, es decir, tal como una cola.

En el siguiente ejemplo se explican los módulos señalados mediante la implementación de éstos.

Ejemplo 6.9

■ Descripción

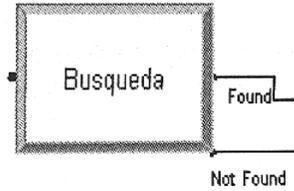
Considere el ejemplo 6.4. Las entidades que quedaron en cola, pues no cumplieron con la condición del diámetro <15 cm, se deberán remover en su totalidad para que sean llevadas a una operación de maquinado, en la cual se le disminuirá el diámetro a la medida requerida; luego se devolverán a la zona de espera para su posterior procesamiento. La operación de maquinado se realiza en un tiempo uniforme entre 5 y 10 minutos; el diámetro queda, entonces, reducido a 13 cm.

Módulos empleados

- Panel de procesos básicos
Módulos lógicos: *Create, Assign, Decide, Process, Dispose*
Módulos de datos: *Entity, Resource, Variables*
- Panel de procesos avanzados
Módulos lógicos: *Hold, Search, Pickup, Dropoff*

■ Desarrollo

El modelo actual comienza con la condición terminal del anterior ejemplo, que consiste en no encontrar ninguna entidad que cumpla con la condición especificada. En esta situación la entidad de control tomará el camino *Not found* del módulo *Search*.



Cuando la entidad de control toma este camino, se deben remover las entidades que quedan en la cola de espera

Primero se debe colocar un módulo *Pickup* para que recoja todas las entidades de la cola.



Pickup

En este campo se coloca la cantidad de entidades que se van a recoger. Si se desean recoger todas en fila, se coloca la variable NQ (*Nombre de la cola*), la cual devuelve el número de entidades en la cola especificada

Posición desde la cual se comienzan a recoger las entidades

Nombre de la cola de la cual se remueven las entidades

A continuación se coloca el módulo *Dropoff*, el cual se utiliza para descargar las entidades del grupo formado por la entidad de control:



Dropoff

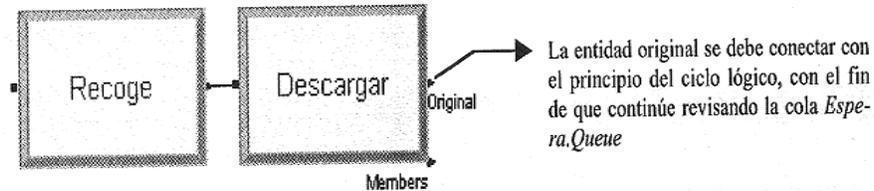
En este campo se coloca la cantidad de entidades que se van a descargar. Si se desean descargar todas las que conforman el grupo, se coloca la variable NG (*Number in group*)

Tratamiento de los atributos asignados a las entidades agrupadas. (Ver módulo *Separate*)

Posición inicial desde la cual comienza la descarga

No necesariamente el módulo *Dropoff* debe ir inmediatamente después del módulo *Pick-up*; esto quiere decir que a partir del momento en el que se recogen y descargan las entidades pueden presentarse distintas clases de módulos que representan demoras, procesos, transportes, etc.

En el módulo *Dropoff* se presentan dos salidas: la de la entidad de control (*Original*) y la de las entidades descargadas (*Members*). La primera tiene prioridad bajo las mismas reglas enunciadas para el módulo *Remove*.



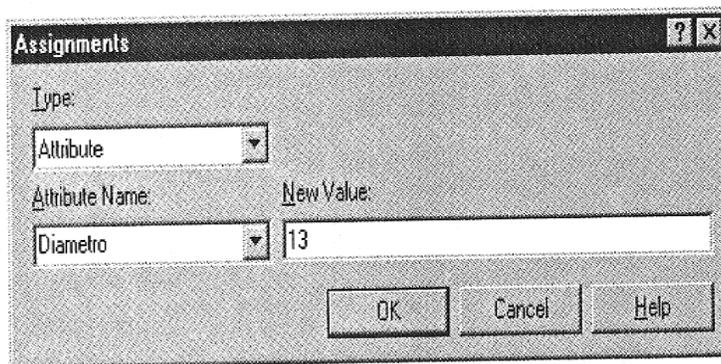
La entidad de control se debe devolver al módulo *Hold (Scan for condition)*, que se conoce como *cola ok*, en el cual continuará verificando y examinando la cola *Espera.Queue*; de esta manera se repite el ciclo.

Por otra parte, las entidades descargadas se pueden enviar a un módulo *Process* con la acción configurada como *delay*, o a un módulo *Delay* del Panel de procesos avanzados para que experimenten el tiempo de procesamiento indicado. En este caso se usará un módulo *Process*.



Process

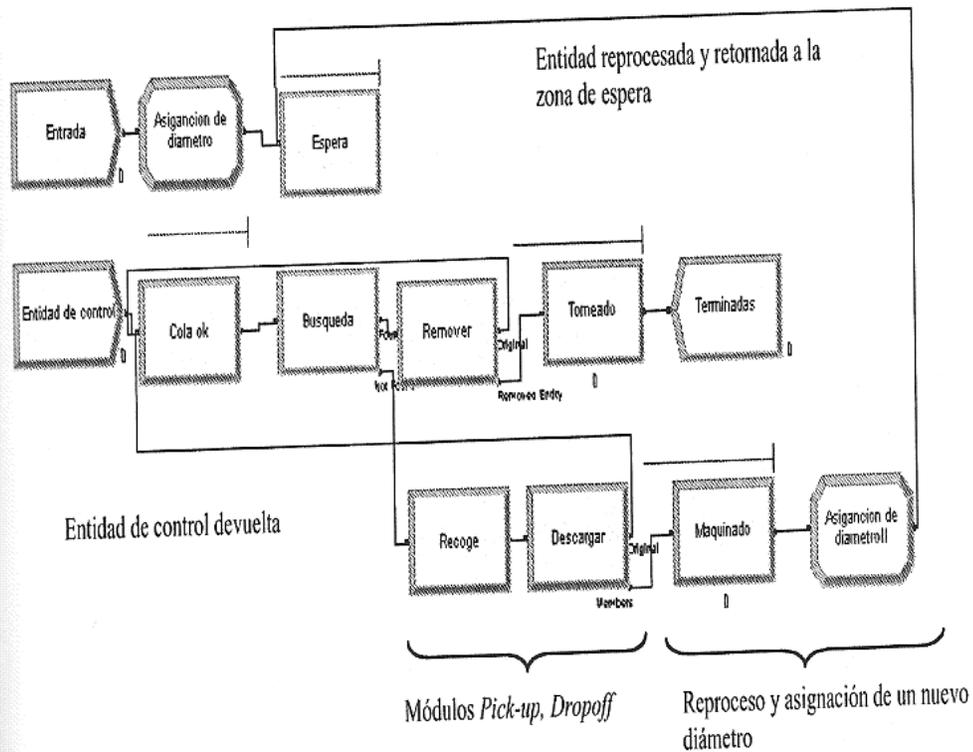
Se debe modificar el valor del atributo *diámetro* mediante un módulo *Assign*, ya que éste es el objetivo de la operación y así se podrá pasar a la inspección inicial.



Se cambia el valor del atributo *diámetro* por una constante según las condiciones que se especifican en el enunciado

Cuando las entidades salen con el nuevo valor del atributo *diámetro*, se envían al módulo *Hold*, que se conoce como *Espera*, con el fin de que sean examinadas (por la entidad de control) e incorporadas al proceso normal.

A continuación se muestra un diagrama del modelo completo en ARENA®:



10. ALTERNATIVAS DE USOS

Mediante la combinación *Search-Pickup* se puede inspeccionar una cola recogiendo entidades que cumplan con una condición específica. Cuando ya no se encuentren más entidades que cumplan esa condición, el valor de la variable *Global J* será 0, y la cola se deja de inspeccionar.

Ejemplo. Un ascensor que se detiene en los pisos 4, 6 y 8 recogerá sólo pasajeros cuyo destino cumpla esta condición.

Mediante la combinación *Search-Dropoff* se puede inspeccionar el total del grupo formado (NG), con el fin de buscar y depositar entidades que cumplan con una condición específica. Nuevamente, cuando ya no se encuentren más entidades que cumplan esa condición, el valor de la variable *Global J* será 0, y el grupo se deja de inspeccionar.

Ejemplo. Cuando un ascensor llega a un piso determinado, se bajan de él sólo las personas cuyo destino sea ese piso.

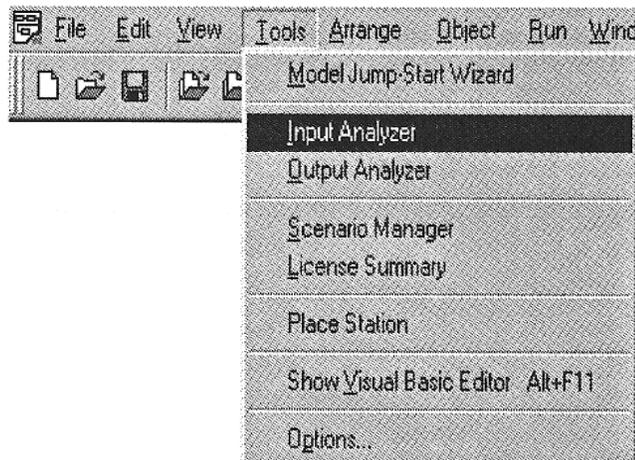
ANALIZADOR DE DATOS DE ENTRADA

El Analizador de datos de entrada o *Input Analyzer* es una poderosa herramienta que se encuentra en el ambiente ARENA[®]. Se puede utilizar para determinar qué distribución de probabilidad se ajusta a los datos de entrada; también para ajustar una distribución específica a los datos, con el fin de comparar funciones de distribución o de visualizar los efectos de cambios en los parámetros de una misma distribución. Además, el *Input Analyzer* puede generar grupos de números aleatorios que se pueden analizar a través de la función de ajuste del programa.

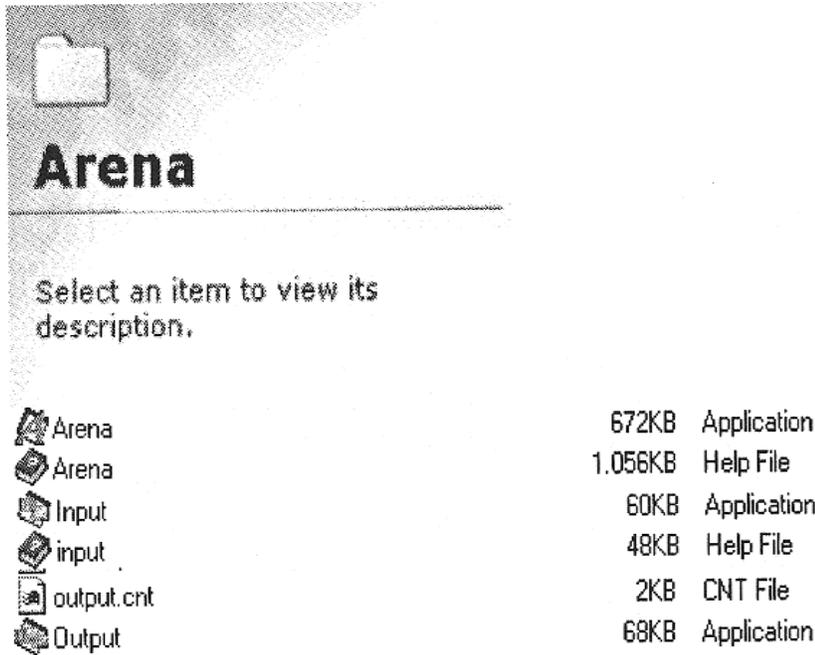
Los datos procesados en el *Input Analyzer*, por lo general, representan intervalos de tiempo asociado con un evento de duración aleatoria. Por ejemplo, se puede usar esta herramienta para analizar un conjunto de datos que representen los tiempos entre llegadas, el tiempo de proceso o el tiempo entre fallas sucesivas del sistema, entre otros aspectos.

1. ENTRADA AL *INPUT ANALYZER*

En ARENA[®] se puede activar esta herramienta desde la barra de menús seleccionando *Tools* y, posteriormente, *Input Analyzer*, así como se muestra a continuación:



Si para activar esta herramienta no se desea abrir ARENA[®], desde el menú de inicio de WINDOWS se puede realizar esta acción, ya que se encuentra en la misma carpeta en la que está el programa:



2. PREPARACIÓN DE LOS ARCHIVOS DE DATOS

Los datos con los que se trabaja en el *Input Analyzer* provienen generalmente de mediciones de tiempo, por ello se deben recopilar en un archivo electrónico. Este archivo se puede crear ingresando los datos a una hoja de cálculo (*Excel*, por ejemplo) o a un procesador de texto (*MS, Word, Wordpad, Notepad*) en ciertas condiciones que se explican posteriormente.

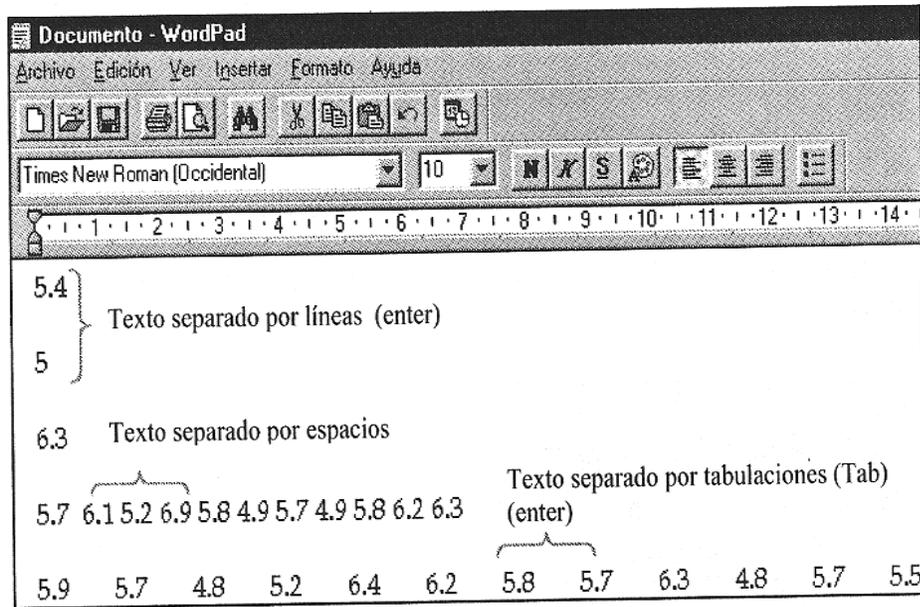
Suponga que los siguientes datos son el resultado de un estudio de tiempos que se realiza en un banco, y que representan el tiempo de atención de un cliente en una caja determinada.

5.4	5.0	6.3	5.7	6.1	5.2	6.9	5.8	4.9	5.7	4.9	5.8	6.2
6.3	5.9	5.7	4.8	5.2	6.4	6.2	5.8	5.7	6.3	4.8	5.7	5.5

Se desea saber qué tipo de distribución de probabilidad se ajusta mejor a estos datos. Esto es particularmente necesario si se va a incluir como un tiempo de proceso en un modelo de simulación.

2.1 PROCESADOR DE TEXTO

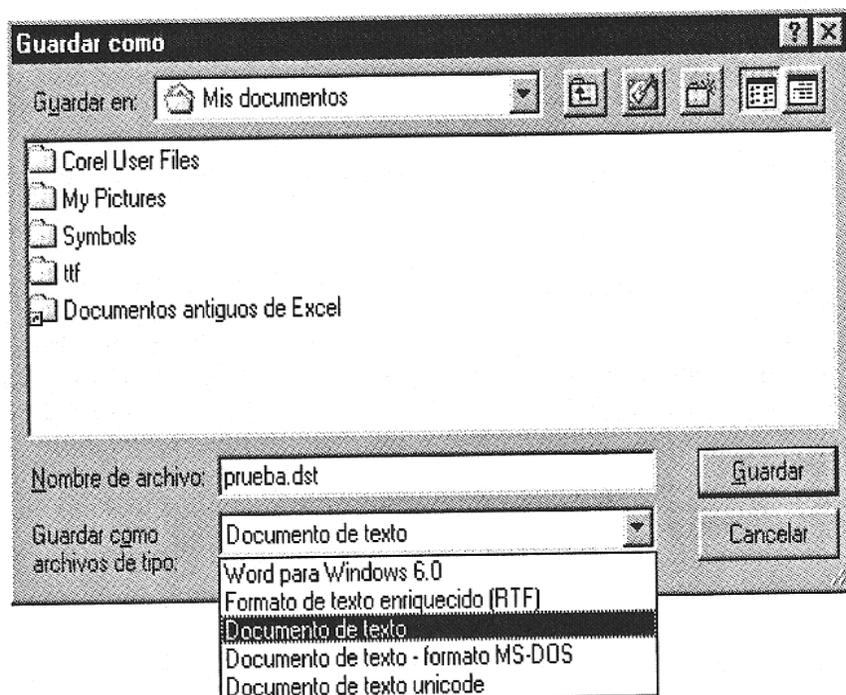
Se puede trabajar en *MS, Word* o *Wordpad*. Para este caso en particular se eligió el segundo. Cuando se trabaja en procesador de texto se deben separar los datos mediante uno o más caracteres de espacio en blanco, como espacios, tabulaciones o líneas.



Luego de ingresar los datos según el formato elegido, se procede a dar al archivo el nombre y extensión requeridos. Los requerimientos del *Input Analyzer* para reconocer un archivo de datos son los siguientes:

- Extensión.dst
- Archivo de sólo texto

El archivo se guarda con el nombre *prueba*. Para efectos de reconocimiento en el *Input Analyzer*, en el cuadro de diálogo *Nombre de archivo* se coloca *prueba.dst* y luego se selecciona la opción *Sólo texto*. A continuación se muestra este procedimiento:

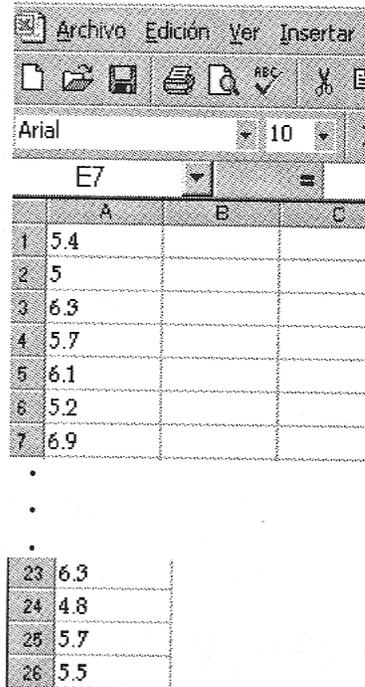


Una vez se termina el anterior procedimiento, el archivo se ve en el explorador de WINDOWS de la siguiente manera:



2.2 HOJA DE CÁLCULO

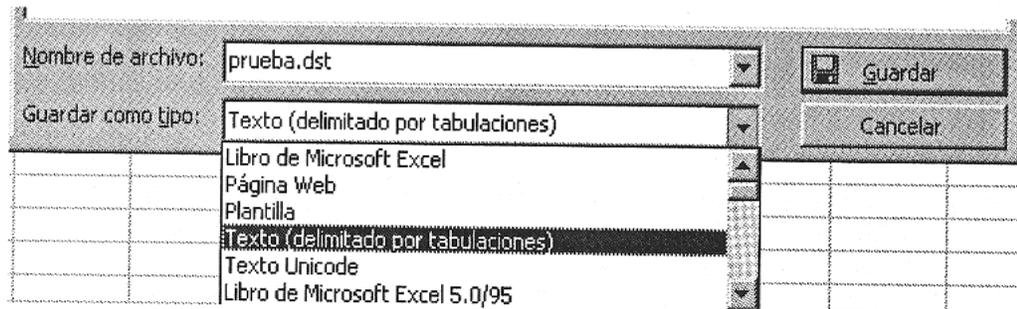
En una hoja de cálculo de *Excel* se ingresan los datos tomados de la manera como se indica a continuación:



The screenshot shows an Excel spreadsheet with the following data in column A:

	A	B	C
1	5.4		
2	5		
3	6.3		
4	5.7		
5	6.1		
6	5.2		
7	6.9		
...			
...			
...			
23	6.3		
24	4.8		
25	5.7		
26	5.5		

Posteriormente se guarda el respectivo archivo según los requerimientos del *Input Analyzer*:



Teniendo en cuenta que los requerimientos del *Input Analyzer* no distinguen si el archivo fue creado en un procesador de texto o en una hoja de cálculo, el método que se

emplea para este propósito es irrelevante; por ello su escongenia se deja a consideración del usuario.

Recomendaciones

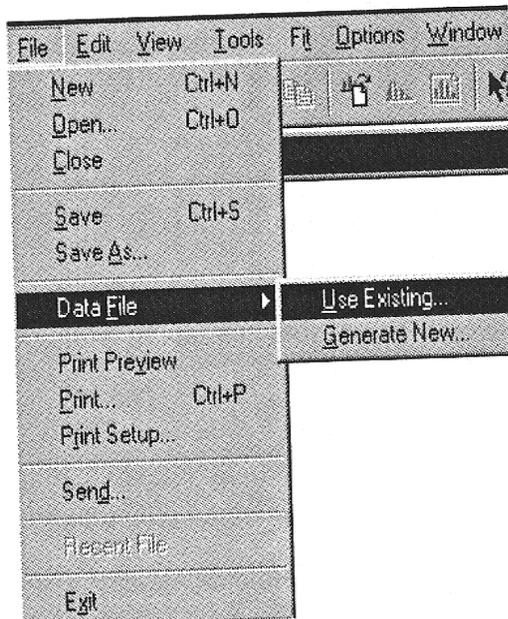
- Los archivos que se generen, ya sea en hoja de cálculo o en procesador de texto, no deben tener ningún tipo de encabezamiento.
- Las porciones decimales se deben separar por puntos (no por comas).
- El archivo no debe tener ningún tipo de carácter diferente a números y puntos decimales.

3. CREACIÓN DE UN ARCHIVO DE ENTRADA (*INPUT FILE*)

Para crear un archivo de entrada se selecciona, de la barra de menús de la ventana del *Input Analyzer*, la opción *File/New* o se utiliza el teclado *ctrl.+n* o se da clic en el icono de nuevo de la barra estándar



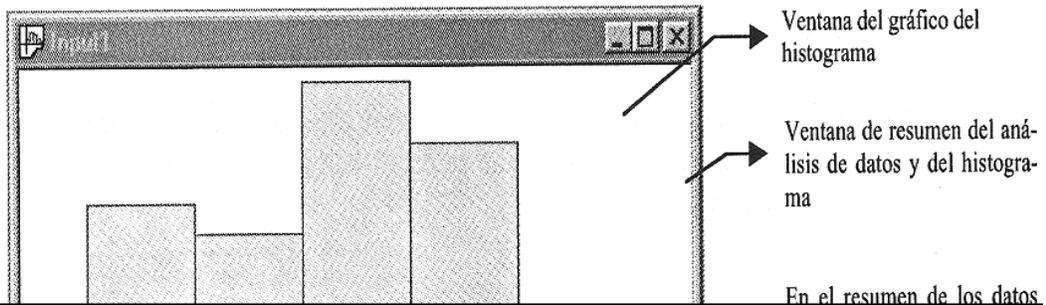
Posteriormente se buscan los datos de origen, es decir, el archivo en el cual se encuentran los datos que se desean analizar. Para esto se selecciona *File/Data File/Use Existing* en la barra de menús; de esta manera se acceden a archivos ya existentes de extensión **.dst*.

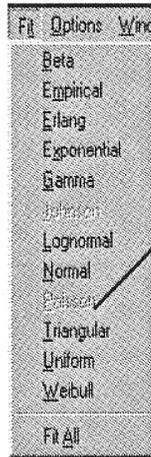


Búsqueda de datos que provienen de un archivo ya creado, con el fin de que sean ajustados

La anterior acción permite que se despliegue un cuadro de abrir archivo; en éste el usuario busca la carpeta en la que se encuentra el archivo de entrada creado. Este cuadro debe mostrar todos los archivos de texto con extensión *.dst*; sin embargo, esto no ocurre si la carpeta en cuestión contiene diferentes tipos de archivos. Para ver los archivos deseados se selecciona, en tipo de archivo, la opción *all files(*.*)* o *text files (*.txt)*.

Al abrir el archivo de interés, en este caso *prueba.dst*, el *Input Analyzer* muestra automáticamente el histograma de los datos tanto gráfica como analíticamente.

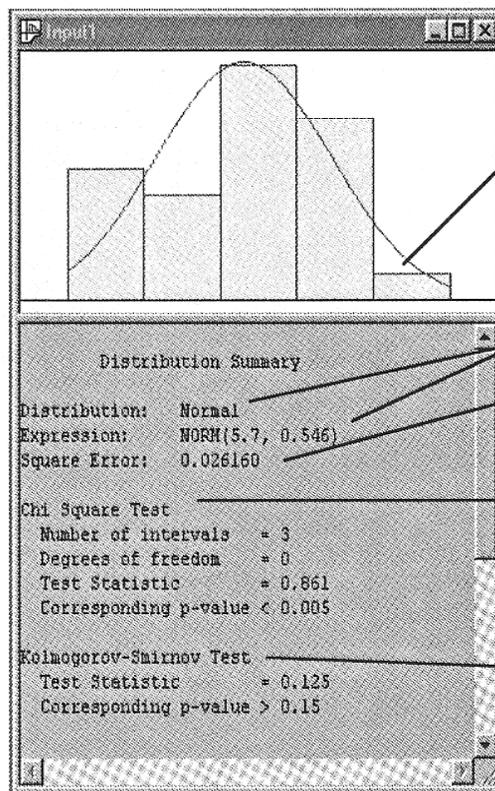




Diferentes opciones de ajuste. Note que la distribución de *Poisson* no está activada; esto ocurre solamente cuando se detectan valores enteros

Posteriormente se escoge la opción para ajustar una distribución normal.

Luego de seleccionar la opción de ajuste deseada, la ventana cambia; aparece en el histograma la curva de ajuste y, en la parte analítica, se muestran los parámetros de la distribución.



Curva de ajuste para una distribución normal

Tipo de distribución y parámetros

Error cuadrado del ajuste

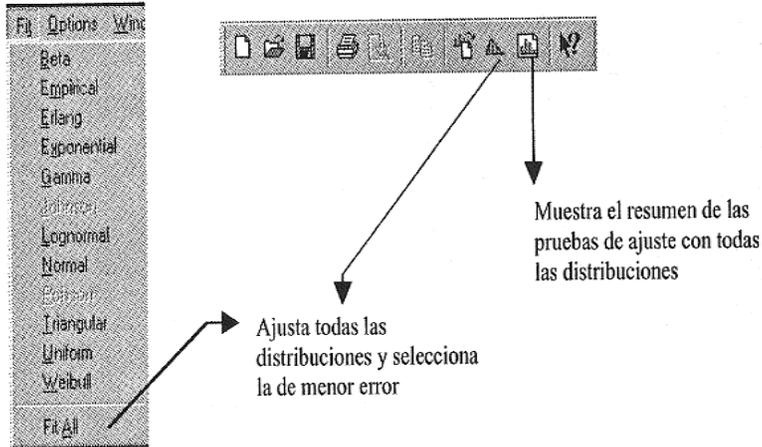
Resultado de la prueba de bondad de ajuste *Chi* con el valor del estadístico p

Resultado de la prueba de bondad de ajuste *Kolmogorov-smirnov* con el valor del estadístico p

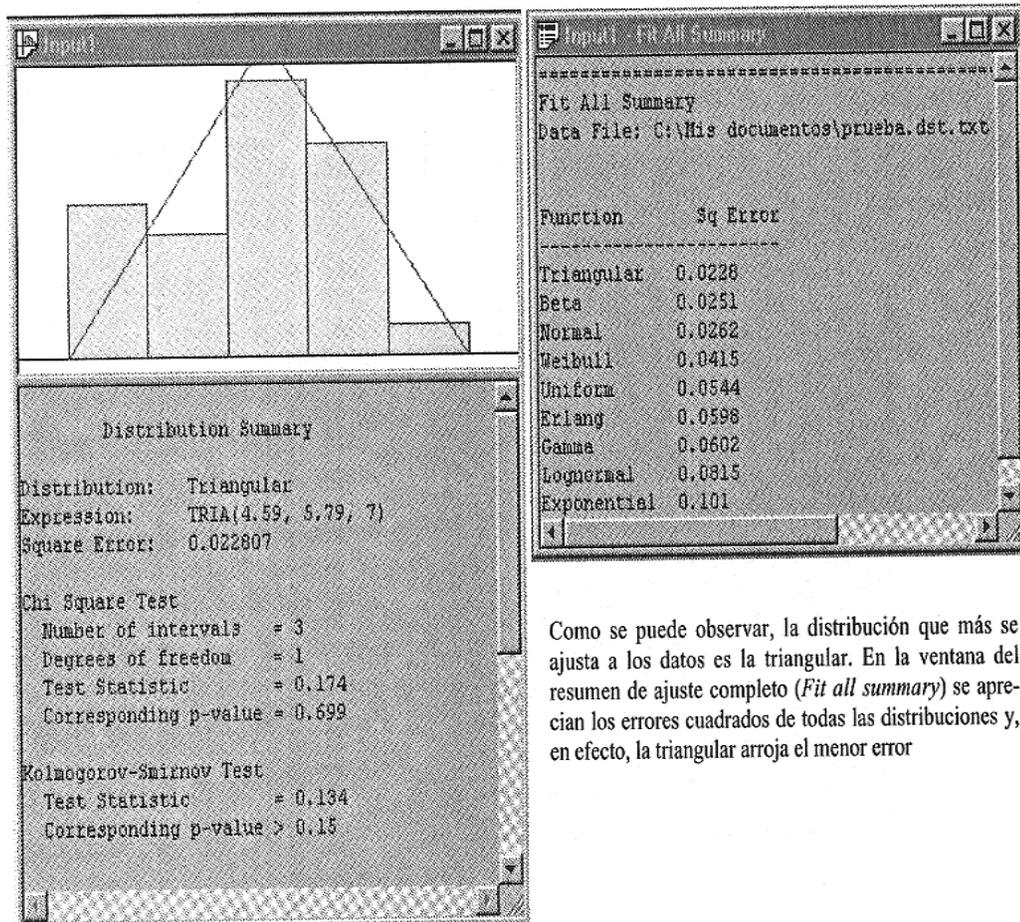
SELECCIÓN DEL MEJOR AJUSTE

El *Input Analyzer* tiene la opción *Fit All*, que permite ajustar todas las distribuciones con que cuenta esta herramienta a los datos analizados, así como seleccionar interna-

mente aquella distribución que arroje un menor error en las pruebas de bondad de ajuste. El resumen de los errores de las pruebas de ajuste con cada una de las distribuciones se muestra haciendo clic en el botón *Fit all summary* de la barra estándar.



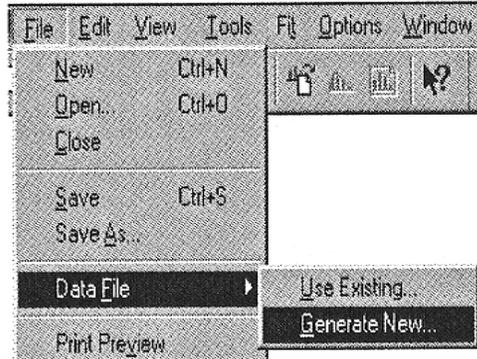
A continuación se presenta la ventana de resultados de la opción de ajuste completo (*Fit All*).



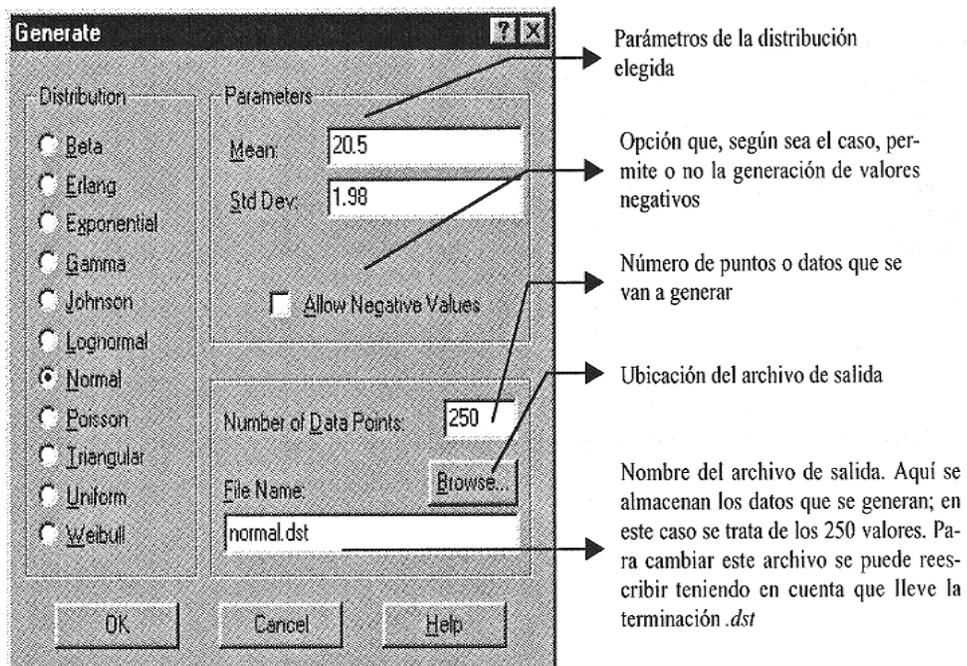
Como se puede observar, la distribución que más se ajusta a los datos es la triangular. En la ventana del resumen de ajuste completo (*Fit all summary*) se aprecian los errores cuadrados de todas las distribuciones y, en efecto, la triangular arroja el menor error

5. GENERACIÓN DE DATOS

Se pueden crear archivos de datos artificiales, es decir, que no provengan de la práctica sino que sean generados automáticamente por el *Input Analyzer*. Esta operación se realiza creando un nuevo archivo de entrada y seleccionando *Generate New* en la opción *Data File*.

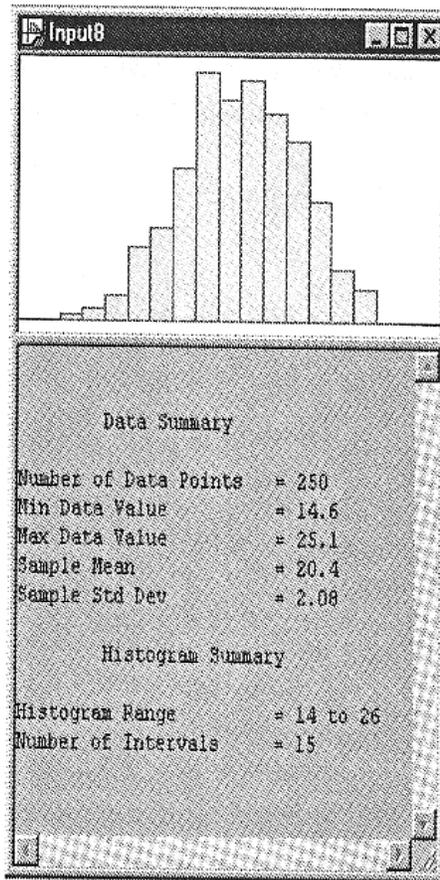


A continuación se muestra un cuadro de diálogo en el cual se puede escoger el tipo de distribución que se desea generar, así como los parámetros. A manera de ejemplo, se generan 250 datos que siguen una distribución normal con media de 20.5 y desviación estándar de 1.98.



Para ver los datos generados, se busca el archivo de salida, que en este caso es *normal.dst*, y se abre en un programa que tenga la capacidad de visualizarlo; este programa puede ser *Wordpad* o *Excel*.

A continuación se muestra la ventana del *Input Analyzer* para los datos generados:



Histograma de los 250 datos generados

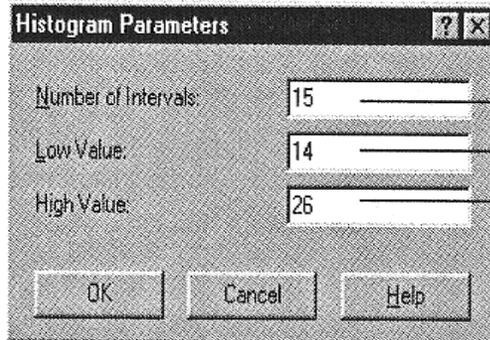
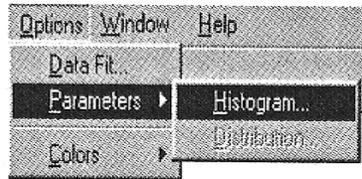
Resumen del análisis de los datos generados

6. OPCIONES DEL HISTOGRAMA

Algunas características del histograma que suministra el *Input Analyzer*, como el número de intervalos, el límite inferior y el límite superior, se pueden variar de acuerdo con ciertas condiciones que se muestran a continuación:

Características del histograma	Condición
Número de intervalos	Debe estar entre 5 y 40
Límite inferior	Debe ser mayor o igual al entero más grande que no sobrepase el menor valor de los datos analizados.
Límite superior	Debe ser menor o igual al entero más pequeño que iguale o exceda el mayor valor de los datos analizados.

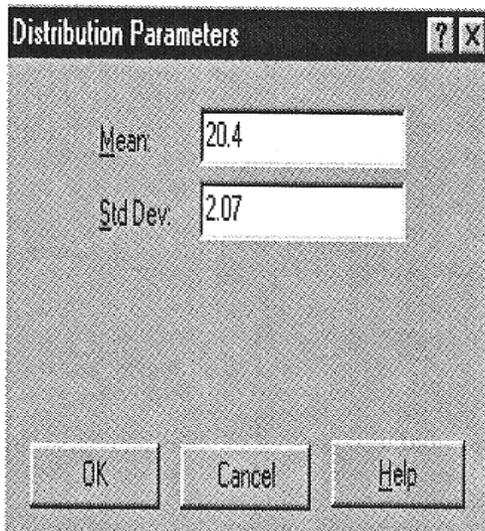
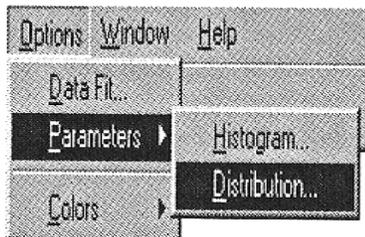
Estos cambios se realizan seleccionando, en la barra de menús, *Options/Parameters/Histogram*; luego se muestra un cuadro de diálogo con los parámetros actuales de la distribución:



Número de intervalos
Límite inferior
Límite superior

6.1 CAMBIOS EN LA DISTRIBUCIÓN

Los parámetros de la distribución que ha sido ajustada a los datos se pueden cambiar seleccionando, en la barra de menús, *Options/Parameters/Distribution*. Si los parámetros de una distribución se cambian, se debe realizar de nuevo la prueba de bondad de ajuste.

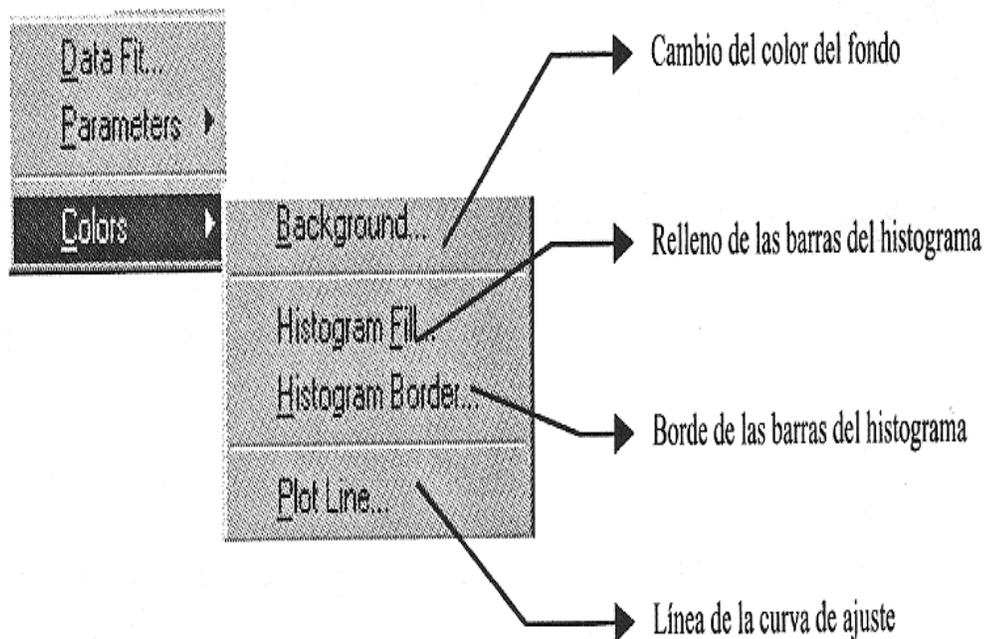


Cuadro de diálogo para cambio de parámetros. En este caso corresponden a una distribución normal
La opción de cambio de parámetro no se activa a menos que una distribución se haya ajustado a los datos

6.2 CAMBIOS EN LA PRESENTACIÓN

Es posible modificar los colores del histograma para mejorar su presentación en el momento de imprimir. Cabe anotar que a veces resulta conveniente llevar la gráfica del histograma a un documento o a una hoja de cálculo. Para esto se hace clic en la gráfica; luego ésta se almacena en el portapapeles mediante las teclas *ctrl.+c*, y se pega en el programa que se necesite.

En la barra de menús se selecciona *Options/Colors*, así como la parte del gráfico que se quiere editar.



Dentro de cada opción se presenta una paleta de colores; en ésta se selecciona el color que se desee colocar en una parte específica del histograma.

PANEL DE TRANSFERENCIA AVANZADA

En el modelamiento de sistemas productivos es frecuente encontrar que las transferencias de material se realizan a través de un elemento transportador. A continuación se mencionan ejemplos de esta situación:

- A través de un dispositivo manual, un operario transfiere un lote de piezas de una estación a otra.
- Los clientes de un complejo turístico pueden ser transportados desde un punto de entrada hacia diferentes sitios, y luego son recogidos y llevados devuelta a la salida.
- El sistema de distribución de productos en una zona geográfica.

Los transportes representan una limitación para la transferencia de entidades, ya que poseen una capacidad finita y un costo asociado. Este aspecto convierte al transporte en un recurso más del proceso productivo; por ello se debe estudiar cuidadosamente.

1. FUNCIONAMIENTO DE UN TRANSPORTE

Una entidad susceptible de ser transferida solicita un transporte; si éste se encuentra disponible, y de acuerdo con sus características (velocidad), se dirigirá hacia la estación en donde se encuentra la entidad solicitante. Una vez el transporte se encuentra en esta estación, la entidad que se va a transferir se carga y se mueve hacia la estación de destino. Al llegar, según sea el caso, puede liberar al elemento transportador para que realice otras funciones o puede seguir en control de éste y mantenerlo reservado en la estación de destino para que siga siendo transferida por el mismo elemento.

Es posible que se cuente con más de un elemento de transporte (flota); si éste es el caso, se puede solicitar una unidad específica. La asignación del transporte se puede realizar de acuerdo con ciertas reglas, a saber:

Cyclical. Asigna de manera cíclica los elementos transportadores de una flota a las entidades solicitantes. Es decir, existe un orden en la flota; el que se usa se coloca en la última posición para que sea asignado.

Random. La asignación se hace de manera aleatoria.

Preferred order. Primero selecciona la unidad de transporte que tenga el orden más bajo dentro de la flota.

Specific members. Solicita un miembro específico o grupo de elementos de la flota.

Largest distance. Solicita el elemento de transporte que se encuentra más lejano de la estación en donde la entidad requiere este servicio.

Smallest distance. Solicita el elemento de transporte que se encuentra más cerca de la estación en donde la entidad requiere este servicio.

2. SOLICITUD DE TRANSPORTE

Un transporte se solicita mediante el módulo *Request*. Éste se encarga de asignar el transporte a la entidad solicitante y mover el elemento transportador hacia la estación en donde ésta se encuentra.



Request

Campo	Descripción
<i>Name</i>	Nombre del módulo. Debe ser único.
<i>Transporter name</i>	Nombre o identificación del transporte o flota.
<i>Selection rule</i>	Regla de sección de transporte. Ésta se explica en apartados anteriores.
<i>Save attribute</i>	Nombre del atributo en el cual se almacena (si se desea) el número del elemento transportador asignado, es decir, su índice en la flota o grupo.
<i>Unit number</i>	Sólo se activa cuando la regla de selección es <i>Specific member</i> . El número de la unidad o elemento de transporte que se solicita se debe ingresar dentro de la flota.
<i>Priority</i>	Le asigna prioridad a la solicitud de transporte para el módulo que se especifique. Es de utilidad cuando se tienen varias solicitudes para un mismo transporte, y una o varias son de más importancia.
<i>Velocity</i>	Velocidad en que se desplazará el vehículo desde el lugar donde éste se encuentra hasta la estación en que se solicita el servicio. (Velocidad sin carga).
<i>Time units</i>	Unidades en que se expresa la velocidad. Se debe tener cuidado de mantener la congruencia en las unidades de distancia y velocidad.
<i>Queue type</i>	Tipo de cola en la cual se almacena la entidad que espera el transporte.
<i>Queue name</i>	Nombre de la cola en la cual la entidad que espera el transporte es temporalmente almacenada. Adopta automáticamente el nombre del módulo.

Dos eventos han ocurrido cuando una entidad que solicita un transporte sale de un módulo *Request*:

- Se le ha asignado un transporte a la entidad solicitante (la capacidad de la flota disminuye en 1).
- El transporte ha llegado a la estación en donde se encuentra la entidad que lo requiere.

Esto no quiere decir que los eventos ocurren inmediatamente después que se inicia la transferencia; ocurren sólo cuando la entidad (que ahora controla el transporte) ingresa en un módulo *Transport*. Para explicarlo de otra manera se puede decir que entre un módulo *Request* y un módulo *Transport* pueden ocurrir diferentes eventos o actividades que se desarrollan sobre la entidad en control del transporte, por ejemplo, carga.



Transport

Campo	Descripción
<i>Name</i>	Nombre del módulo. Debe ser único.
<i>Transporter name</i>	Nombre o identificación del transporte o flota.
<i>Unit number</i>	Este campo se llena sólo cuando se solicita una unidad específica dentro de la flota.
<i>Destination type</i>	Se especifica el lugar hacia donde se dirige la entidad; puede ser hacia una estación (<i>Station</i>). Se puede usar una secuencia (<i>Sequential</i>); así mismo, la información de su destino puede estar almacenada en un atributo (<i>Attribute</i>) o en la evaluación de una expresión (<i>Expression</i>).
<i>Velocity</i>	Velocidad del elemento que se expresa en unidades de longitud y que es equivalente a la distancia que recorre el transporte (velocidad con carga).
<i>Time units</i>	Unidades de tiempo para la velocidad.

Cuando una entidad llega a la estación de destino, ésta sigue teniendo control del transporte hasta cuando se le dé la instrucción de liberarlo mediante el módulo *Free*. Análogamente al conjunto *Request-Transport*, entre un módulo *Transport* (no están unidos) y un módulo *Free* (con un módulo *Station* antes) pueden ocurrir diferentes eventos o actividades que se desarrollan sobre la entidad en control del transporte en la estación de destino. Ejemplo, descarga.



Free

Campo	Descripción
<i>Name</i>	Nombre del módulo. Debe ser único.
<i>Transporter name</i>	Nombre de la flota a la cual pertenece el transporte.

El movimiento de un transporte está determinado por la distancia entre estaciones y por la velocidad que se especifique (con carga o sin carga). Las distancias y las características del transporte se especifican en los módulos de datos *Distances* y *Transporter*, respectivamente.

Ejemplo 8.1

■ Descripción

Considere que en el caso del ejemplo 2.6, las partes se trasladan en 2 unidades de transporte, las cuales se encargarán de la transferencia de entidades desde la entrada hasta la estación A y la estación B.

Considere que existe un tiempo de carga y descarga igual en todas las estaciones, que es constante a 0.2 minutos. Las velocidades con carga y las distancias entre estaciones se muestran a continuación:

Estaciones	Distancia	Velocidad
Entrada-Estación A	10 m	2
Estación A-Estación B	10 m	0.5
Entrada-Estación B	20 m	-

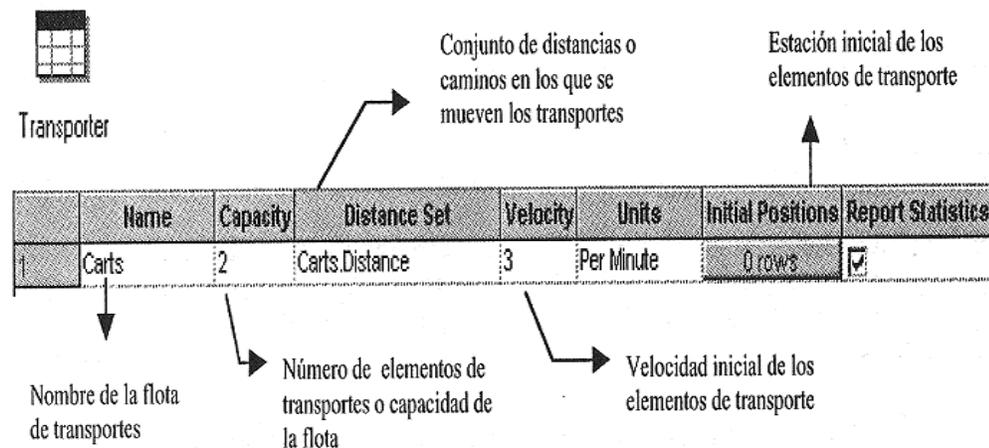
Es preciso que se definan caminos entre todas las estaciones que visita el transporte, ya que si falta uno, el programa mostrará un error en tiempo de corrida. En el ejemplo en estudio, si no se coloca el camino desde la estación de entrada hasta la estación B, cuando una entidad ingrese al sistema y solicite un transporte que esté en la estación B, el transporte no tendrá un camino que le asegure llegar a la entrada, y el modelo se detendría por un error.

La velocidad sin carga del transportador es de 3m/min.

Adicionalmente, suponga que tiene prioridad una solicitud de transporte de la estación de entrada sobre una que proviene de la estación A.

■ Desarrollo

Primero, mediante el módulo de datos *Transporter* se crea el elemento transportador en el panel de transferencia avanzada:



De acuerdo con el módulo de datos de transporte, un elemento transportador es una flota con capacidad 1. En este caso, el grupo completo se ha denominado *Carts* y se le ha asignado una capacidad de 2; es decir, se cuenta con dos elementos de transporte.

Mediante el módulo de datos *Distance* se crean, posteriormente, las distancias o caminos que seguirá el transporte:



Distance

	Name	Stations
1	Carts.Distance	0 rows

Se asignan los pares de estaciones y las distancias entre éstas de todos los caminos a través de los cuales se desplazará el transporte

Cuando se crea un elemento transportador, automáticamente se crea el grupo de distancias o caminos asociados a éste

	Beginning Station	Ending Station	Distance
1	Entrada	Estacion_A	10
2	Estacion_A	Estacion_B	10
3	Entrada	Estacion_B	20

Double-click here to add a new row.

Se adicionan los caminos entre las estaciones a través de las cuales se mueven los transportes, así como las distancias entre estaciones

Luego de crear las características del transporte se procede a agregar el módulo lógico para el problema.

Para solicitar el transporte se adiciona un módulo *Request* a continuación del módulo *Station* que define la estación de entrada:



Request

Request	
Name:	Solicita transporte en entrada
Transporter Name:	Carts
Selection Rule:	Cyclical
Priority:	High(1)
Velocity:	3
Units:	Per Minute
Queue Type:	Queue
Queue Name:	Solicita transporte en entra

OK Cancel Help

Nombre del módulo con la especificación de su función

Transporte o grupo de transportes desde el cual se solicita el elemento transportador

Regla de selección del elemento de transporte

Prioridad sobre otras solicitudes de transportes; en este caso es alta

Velocidad, en distancia y en unidades de tiempo, a la cual se desplazará el dispositivo de transporte solicitado (velocidad sin carga)

Cola en la cual se almacenarán las entidades que esperan por el transporte

Una vez el transporte se encuentra en la estación de entrada, la entidad sale del módulo *Request* e ingresa en el modulo *Delay* para representar la demora por el tiempo de carga.



Delay

Nombre del módulo e indicación de su respectiva función

Demora y unidades de tiempo en que ésta se presenta

Hasta ahora la entidad ha tomado control del transporte; lo ha demorado por 0.2 minutos (tiempo de carga), pero todavía se encuentran (entidad y transporte) en la estación de entrada. Para hacer la transferencia efectiva se agrega un módulo *Transport*:



Transport

Nombre del módulo y especificación de su función

Nombre del transporte que va a realizar la transferencia

Tipo de destino, referenciado por una estación; en este caso se trata de la estación A

Velocidad de transferencia hacia la estación A (velocidad con carga)

A continuación se muestra la secuencia de módulos lógicos:



Estación A

Después de completar la transferencia, la entidad se encuentra en la estación A (módulo *Station*) y todavía controla el transporte. En este punto se adiciona un módulo *Delay* con el fin de representar la demora por descarga:



Delay

Nombre del módulo y especificación de su función

Demora y unidades de tiempo en que ésta se presenta

Delay

Name: Tiempo de descarga en A Allocation: Other

Delay Time: 0.2 Units: Minutes

OK Cancel Help

Posteriormente se procede a la liberación del elemento de transporte; para ello se adiciona un módulo *Free*:



Free

Nombre del módulo y especificación de su función

Nombre del transporte o grupo al cual pertenece el elemento que se libera

Número de la unidad o elemento de transporte que se libera. Se deja en blanco, ya que por omisión el programa libera el que la entidad esté ocupando en ese momento

Free

Name: Libera transporte en A

Transporter Name: Carts Unit Number:

OK Cancel Help

Una vez se libera la entidad, ésta pasa al módulo *Process* respectivo (proceso en A), cuya edición se explica en instancias anteriores.



Process

Después de concluir el procesamiento en el recurso A, la entidad está lista para su transferencia a la estación B, donde se realizará su próxima operación, así que solicita un transporte mediante un módulo *Request*:



Request

The screenshot shows a dialog box titled "Request" with the following fields and annotations:

- Name:** Solicita transporte en A. (Annotation: Nombre del módulo y especificación de su función)
- Transporter Name:** Carts. (Annotation: Transporte o grupo de transportes del cual se solicita el elemento transportador)
- Selection Rule:** Cyclical. (Annotation: Regla de selección del elemento de transporte)
- Save Attribute:** (Empty field)
- Priority:** Low(3). (Annotation: Prioridad sobre otras solicitudes de transportes. En este caso es baja con respecto a la que se hace en la entrada)
- Velocity:** 3. (Annotation: Velocidad, en distancia y en unidades de tiempo, a la que se desplazará el dispositivo de transporte solicitado (velocidad sin carga))
- Units:** Per Minute. (Annotation: Velocidad, en distancia y en unidades de tiempo, a la que se desplazará el dispositivo de transporte solicitado (velocidad sin carga))
- Queue Type:** Queue. (Annotation: Cola en la cual se almacenarán las entidades que esperan por el transporte)
- Queue Name:** Solicita transporte en A.Qu. (Annotation: Cola en la cual se almacenarán las entidades que esperan por el transporte)

Buttons: OK, Cancel, Help.

Cuando se sale del módulo *Request*, y antes de efectuar la transferencia, el tiempo de carga se debe representar mediante un módulo *Delay*:



Delay

Nombre del módulo y especificación de su función

Demora y unidades de tiempo en que ésta se presenta

Luego de experimentar la demora por carga en la estación A, se procede a realizar la transferencia de la entidad. Esta operación se lleva a cabo mediante un módulo *Transport*, así como se muestra a continuación:



Transport

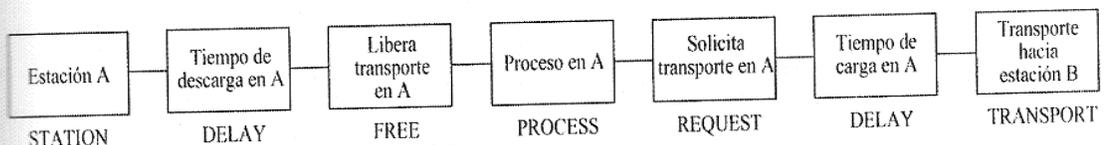
Nombre del módulo y especificación de su función

Nombre del transporte que va a realizar la transferencia

Tipo de destino referenciado por una estación. En este caso se trata de la estación B

Velocidad de transferencia hacia la estación B (velocidad con carga)

A continuación se muestra la secuencia de módulos lógicos:



Estación B

Análogamente al procedimiento empleado en la estación A, en esta operación se debe descargar; por ello la entidad y el transporte deben experimentar una demora. Una vez se está en la estación B, y antes de liberar el transporte, se agrega un módulo *Delay*:



Delay

Nombre del módulo y especificación de su función

Demora y unidades de tiempo en que ésta se presenta

Posteriormente se procede a liberar el transporte mediante un módulo *Free*:



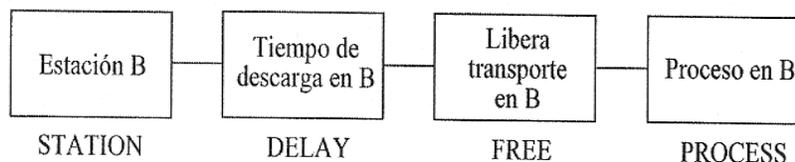
Free

Nombre del módulo y especificación de su función

Nombre del transporte o grupo al cual pertenece el elemento que se libera

Número de la unidad o elemento de transporte que se libera. Se deja en blanco, ya que por omisión el programa libera el que la entidad esté ocupando en ese momento

A continuación se presenta la secuencia de módulos lógicos:



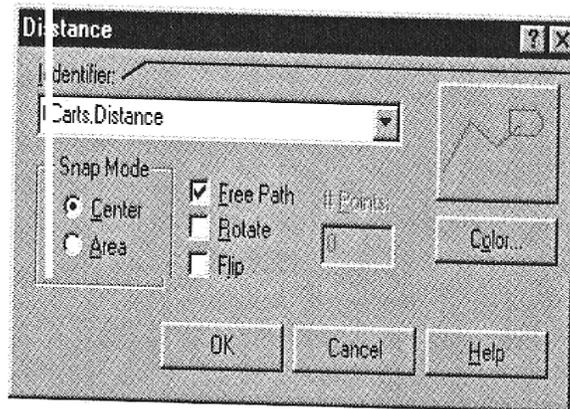
3. ANIMACIÓN

3.1 ANIMACIÓN DEL CAMINO ENTRE ESTACIONES

Como se explica anteriormente, se tienen tres caminos posibles (con una distancia específica) que los elementos transportadores van a seguir; por ello se deben representar en el modelo. Todos los caminos (distancias) asociados al transporte o grupo de transportes *Carts* reciben el nombre de *Carts.Distance*.

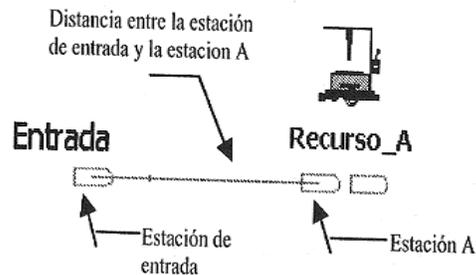
Distancia Entrada-Estación A

Se emplea el conector para distancias *Distance* que se encuentra en la barra *Animate Transfer*:



Conjunto de distancias al cual pertenece la animación que se está creando. Se deben conectar solamente las estaciones definidas en los pares especificados del módulo de datos *Distance*

Cuando se presiona *ok*, el puntero del ratón se transforma en cruz y se unen las estaciones respectivas:



Distancia Estación A-Estación B

De igual manera que en el procedimiento anterior, en esta operación se recurre al conector para distancias *Distance* que se encuentra en la barra *Animate Transfer*:

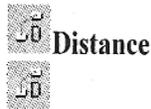


Cuando se especifica el grupo de distancias que se está animado y se presiona *ok* en el cuadro de diálogo de animación de distancia (*Distance*), el puntero del ratón se transforma en cruz y se unen las dos estaciones de interés:

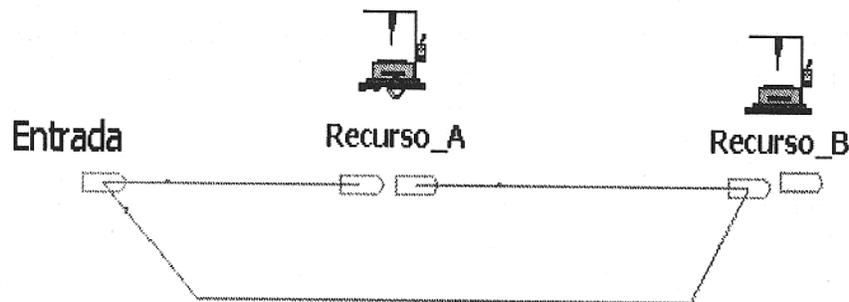


Distancia Entrada-Estación B

Nuevamente se recurre al conector para distancias *Distance* que se encuentra en la barra *Animate Transfer*:



Luego de especificar el grupo de distancias que se está animado y de presionar *ok* en el cuadro de diálogo de animación de distancia (*Distance*), el puntero del ratón se transforma en cruz y, entonces, se procede a unir las dos estaciones de interés:

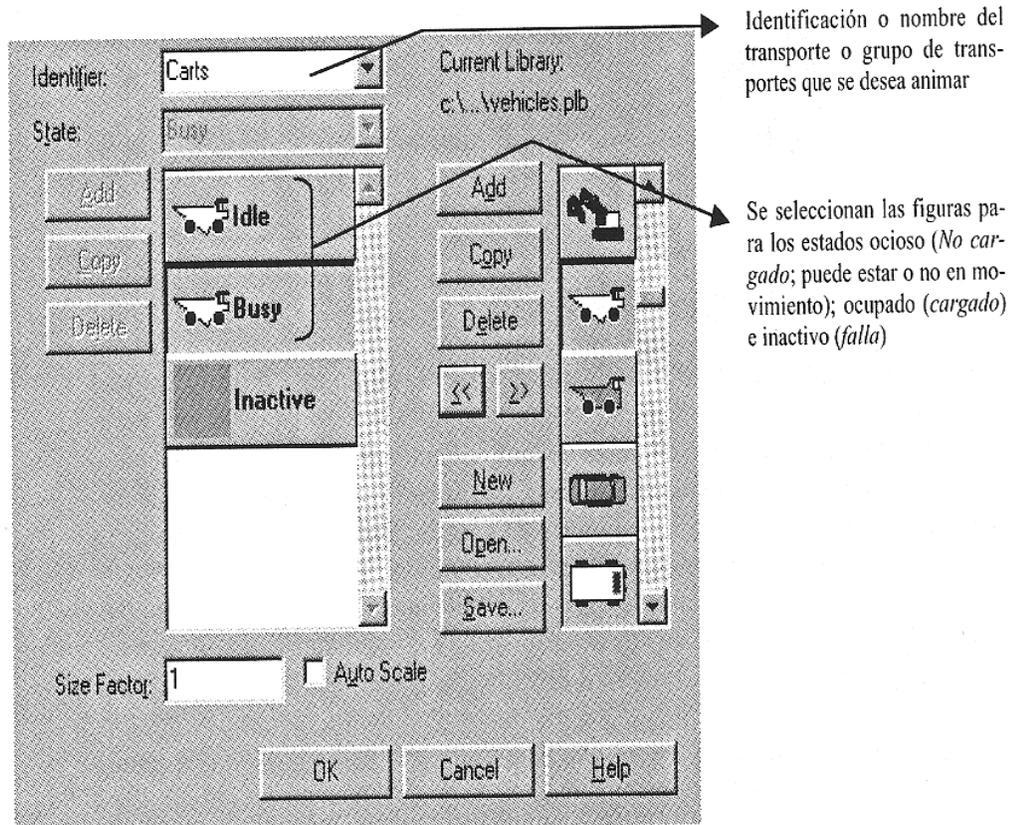


3.2 ANIMACIÓN DEL ELEMENTO TRANSPORTADOR

El dispositivo o medio de transporte empleado se puede representar mediante una figura que, según sea el caso, puede ser montacarga, camión, carro manual, etc. La animación del elemento transportador se define mediante el icono de animación de transporte de la barra *Animate Transfer*:

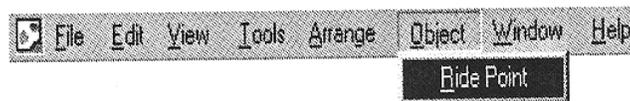
Transporter

La selección de una figura predeterminada se hace de manera análoga a la animación de recursos (ver *Animación* en el capítulo 4). Se selecciona la librería adecuada (*vehicles.plb*) y luego se aloja en el mismo directorio que el ejecutable *Arena.exe*.

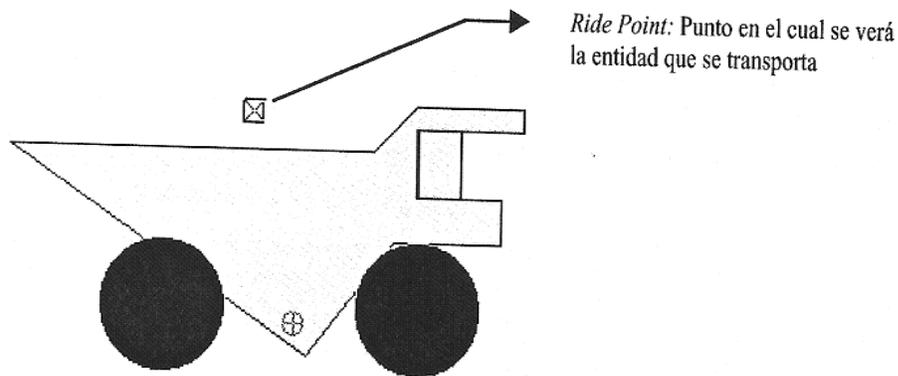


Para constatar que el transporte se encuentra ocupado, es conveniente mostrar la entidad que se está transportando. Con este propósito se hace doble clic en la casilla donde se encuentra la animación de transporte ocupado (*Busy*).

Este procedimiento permite el ingreso al editor de figuras de ARENA[®]. En éste, y a través del menú *Object/Ride Point*, se coloca el punto en el que se desea ver la entidad sobre el elemento de transporte durante la trayectoria del movimiento.



Cuando se selecciona esta opción, el puntero del *mouse* se transforma en cruz; se indica, entonces, en qué lugar de la figura se desea que aparezca la entidad que se está transportando.



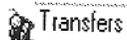
4. CORRIDA Y ANÁLISIS DE RESULTADOS

La versión estudiantil de ARENA[®] recomienda correr el modelo en 1.700 minutos y no en el tiempo que se propone en el enunciado original del ejemplo en estudio, es decir, 2.400 minutos. Para que las estadísticas de los transportes se muestren en los reportes, se debe activar esta opción mediante el menú *Run/Setup/Project parameters* (de la manera como se indica en la sección de bandas de transportadores), así como la casilla de verificación correspondiente a la recolección de estadísticas de transportes:

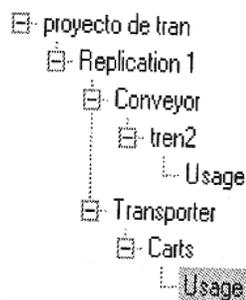


A continuación se muestran los resultados para un modelo combinado. En este caso se utilizan transportes entre la estación de entrada y la estación B, y se usa la banda *tren 2* desde la estación B hasta la bodega:

- Se seleccionan los reportes de transferencias:



- En estos reportes se seleccionan las estadísticas de uso:



Carts	
Usage	Average
Number Busy	1.9980
Number Scheduled	2.0000
Utilization	0.9990

Número promedio de elementos de transporte que se utilizan. En este caso se usan aproximadamente 2 de 2. Este resultado, que cuenta con una alta utilización de la flota (*Utilization*), indica que se necesita más capacidad de transporte

Si la capacidad de la flota se incrementa a 10, los resultados serían los siguientes:

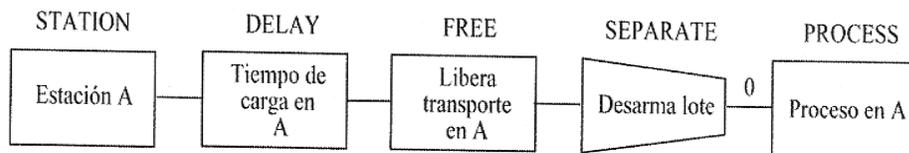
Carts	
Usage	Average
Number Busy	3.6605
Number Scheduled	10.0000
Utilization	0.3660

Los nuevos reportes indican que de una capacidad de 10 sólo se utilizan 3.66 carros; por lo tanto, es posible que con 4 carros se logre un buen desempeño. Esto incrementaría la utilización de la flota

En las condiciones anteriormente enunciadas, un elemento transportador puede transferir sólo una entidad a la vez, lo cual, en primera instancia, podría parecer limitante. Considere, por ejemplo, formar un lote temporal con un tamaño de 10 (módulo *Batch*) antes de que las entidades ingresen al módulo *Request*, así como se muestra a continuación:



Mediante el módulo *Batch* se crea una entidad que representa al grupo o lote formado por 10 entidades. Este lote es transportado y, después de liberar el transporte, separado



El lote se separa antes de ser procesado y después de liberar el transporte

Los resultados se muestran a continuación:

Carts	
Usage	Average
Number Busy	3.8538
Number Scheduled	4.0000
Utilization	0.9635

Adicionalmente, el tiempo de flujo de las entidades (*Total time*) disminuye notablemente.

HERRAMIENTAS DE ANÁLISIS

1. PROCESS ANALYZER / ANALIZADOR DE PROCESOS

La simulación es una herramienta que permite el análisis de sistemas al capturar algo de la complejidad y variabilidad inherentes de la realidad para constituir un soporte robusto en el momento de la toma de decisiones.

Mediante un modelo de simulación se pueden desarrollar los siguientes procesos:

Optimización. Permite determinar la configuración de ciertas variables que maximicen o minimicen una medida de desempeño según sea el caso.

Predicción. Permite estimar el efecto tanto de variaciones en el sistema como del entorno.

Conocimiento. Permite:

- Confirmar relaciones causa-efecto de ciertos eventos.
- Confirmar que todas las variables relevantes son conocidas.
- Evaluar la sensibilidad del modelo ante variaciones en los parámetros más importantes.

Justificación

- Permite decidir entre dos alternativas con base en el valor de una medida de desempeño.
- Un modelo de simulación con una buena animación puede ser un generador de nuevas ideas.

En un sistema se encuentran factores controlables e incontrolables que producen algún efecto en una determinada variable de respuesta. Desde este enfoque se puede emplear la simulación para evaluar el efecto que tendrían ciertos valores de estos factores o variables, tanto controlables como incontrolables, sobre la o las variables de respuesta que se escojan. De esta manera, cada conjunto de valores para los factores antes mencionados recibe el nombre de *escenario*. Cabe resaltar que en la realidad los factores incontrolables no se pueden variar según el criterio del analista; sin embargo, mediante la simulación se pueden estimar los efectos que las variaciones de éstos tendrían sobre el sistema.

ARENA® cuenta con una herramienta llamada *Process Analyzer* (Analizador de procesos), la cual permite al analista generar, evaluar y completar diversos escenarios de un

sistema, previa construcción, verificación y validación del modelo en términos de una o unas variables de respuesta específica.

Ejemplo 9.1

En un sistema de colas en un banco se puede establecer el efecto que tendría incluir un cajero más (factor controlable) en los parámetros de atención o variables de respuesta (tiempo de espera, tiempo en el sistema, número de clientes atendidos por hora, entre otros). Incluso, también se puede estimar el efecto que tendría el incremento en la tasa de llegada de clientes (factor incontrolable) teniendo en cuenta la capacidad actual.

Ejemplo 9.2

En un sistema de inventarios se puede estimar el efecto que tendría un cambio en las políticas (R, Q o S, s) sobre los costos totales de operación. También se puede determinar que tan robusta sería la política actual ante incrementos en las cantidades demandadas (factor incontrolable).

Para ilustrar el uso de la herramienta en estudio se analizará el ejemplo 2.1, en el cual se presenta el caso de un sistema de colas en un banco al que llegan clientes con tiempo entre llegadas que sigue una distribución normal con media de 5.89 y una desviación estándar de 0.65. Los clientes son atendidos por un cajero en un tiempo que sigue una distribución exponencial con media de 4.5.

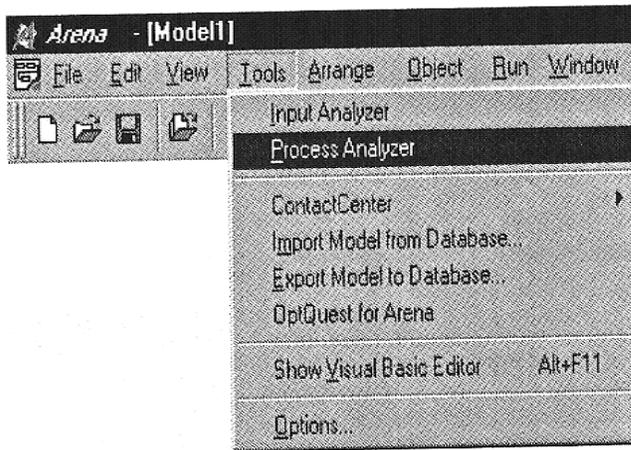
En este caso se tienen las siguientes posibles variables de respuesta del sistema:

- El tiempo que espera un cliente en cola (*Wait time*)
- El tiempo que pasa un cliente en el sistema (*Total time*)
- La longitud promedio de cola (*Number in queue*)
- El número de clientes en el sistema (*WIP*)
- El número promedio de cajeros ocupados (*Number busy*)
- La utilización de los cajeros (*Utilization*)

La variable de control es el número de cajeros. Esta variable está representada por la capacidad del recurso cajero, que se puede ver en la hoja de datos de recursos (*Resources*) del panel de procesos básicos. El objetivo es explorar diferentes escenarios para observar el efecto que tiene, en las variables de respuesta, trabajar con 1, 2, 3 ó 4 cajeros.

Según las indicaciones del ejemplo 2.1, se correrá el modelo durante 480 minutos, y para lograr un mejor análisis se harán 5 réplicas por cada escenario. Una vez se construye y verifica el modelo (se asumirá validez), se ingresa al *Process Analyzer* por medio de la barra de menús *Tools/Process Analyzer* o a través de la ruta *Rockwell Software/Arena/Process Analyzer* en el menú de inicio.

Importante. El modelo que se va a analizar no debe tener errores; debe haber sido guardado y revisado por el programa (tecla F4) para que pueda generar el archivo *p* con el que trabaja el analizador de procesos.

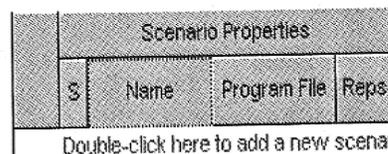


Luego que se esté en el programa se procede a crear un proyecto nuevo mediante la ruta *File/New* o a través del icono de nuevo proyecto:

 Nuevo proyecto

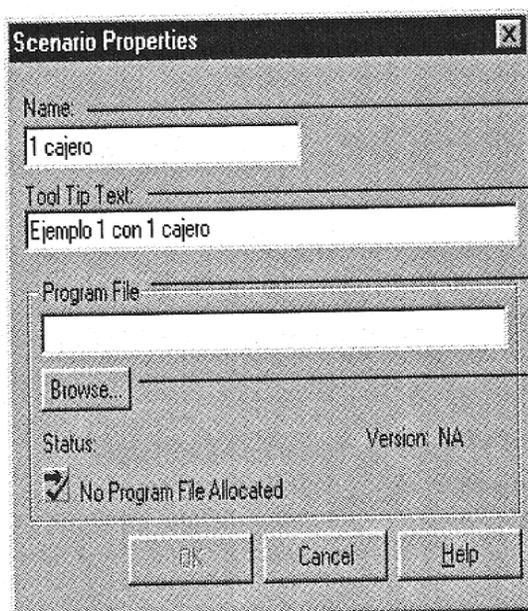
1.1 CREACIÓN DE UN ESCENARIO

En el nuevo proyecto se encuentra la hoja de datos de escenarios.



Se crea el primer escenario haciendo doble clic en el sitio indicado

A continuación se muestra el cuadro de diálogo de las propiedades del escenario que se va a crear:

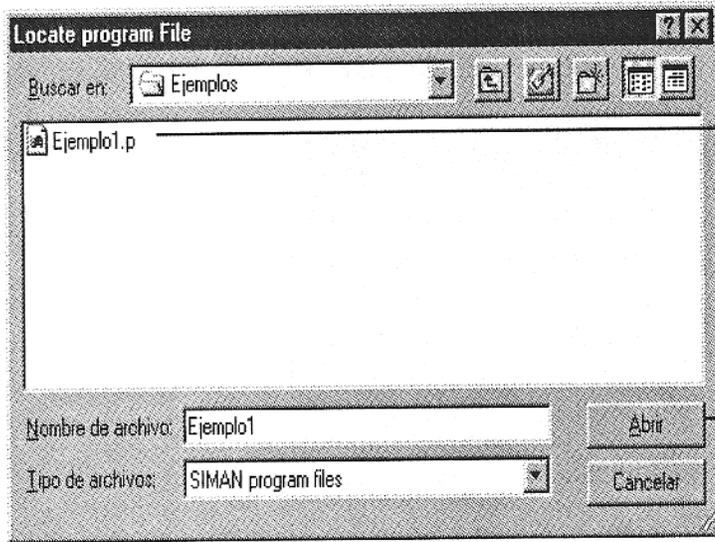


Nombre del escenario que se va a crear

Breve descripción del escenario

Directorio en el cual se encuentra el modelo sobre el cual se va a trabajar

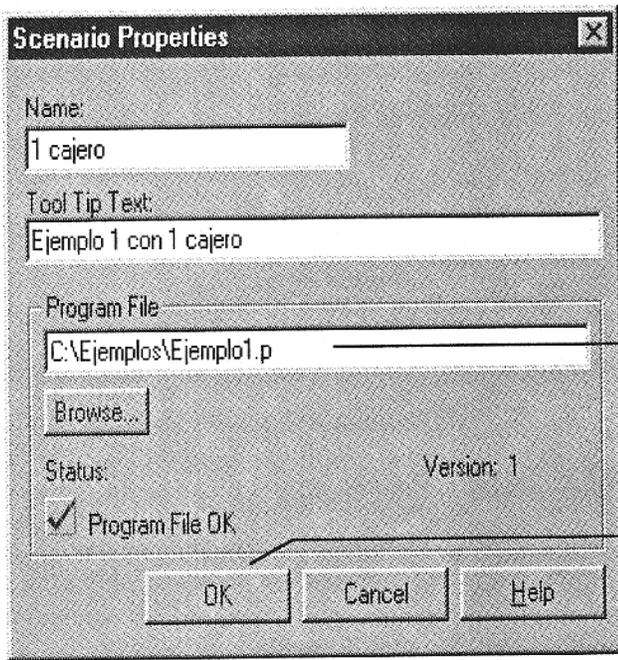
Para buscar el directorio se presiona el botón *Browse*. Se debe buscar el archivo *p*, el cual se genera al revisar o correr un modelo. Este archivo recibe el mismo nombre del modelo (con extensión **.p*) y se genera en el mismo directorio del modelo



El cuadro de diálogo filtra todos los archivos y deja ver sólo los de tipo *p* (*Siman Program File*)

Una vez se localiza el archivo *p* del modelo que se desea analizar, éste se abre para el escenario creado

Después de localizar el programa se retorna al cuadro de diálogo de las propiedades del escenario, en el cual se puede ver la ruta en la que se encuentra el archivo seleccionado.



Directorio del archivo *p* para el modelo del ejemplo 9.1

Con esta operación se define el modelo sobre el cual se realizará el escenario 1. Pulsando *ok* se termina de crear el escenario

Scenario Properties				
S	Name	Program File	Reps	
1	✓ 1 cajero	1: Ejemplo1.p	0	

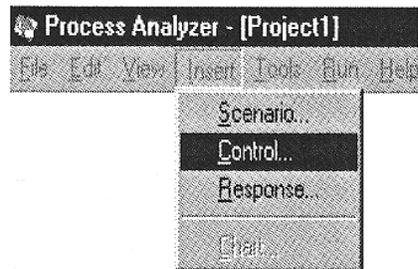
Double-click here to add a new scenario.

Cuando se regresa a la ventana del *Process Analyzer* se tiene creado el escenario 1

Si se van a crear los escenarios sobre el mismo modelo, es conveniente crear un primer escenario especificando todas las variables de respuesta y actores de control, para después duplicarlo y, de esta manera, evitar la repetición de los pasos enunciados.

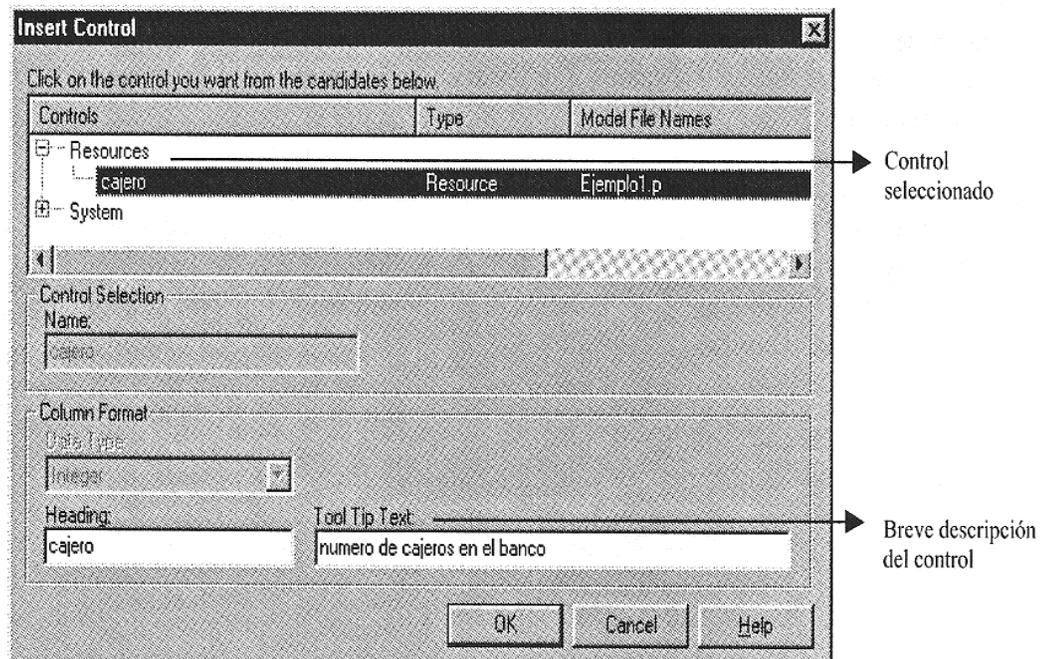
1.2 DEFINICIÓN DE CONTROLES

En el menú se adiciona un factor controlable o simplemente un control.



En el cuadro de adición de controles se muestran todas la variables del modelo que se pueden cambiar desde el *Process Analyzer*.

Para el caso particular del ejemplo en estudio se puede cambiar sólo la capacidad de atención del recurso cajero; así que ése será el control seleccionado.



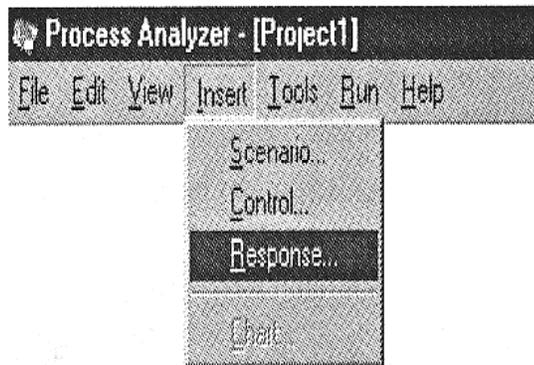
Se puede seleccionar sólo un control a la vez. Cuando se presiona *ok* se vuelve a la ventana del proyecto, en la cual la hoja de datos de los escenarios se actualiza con el control adicionado:

Scenario Properties				Control
S	Name	Program File	Reps	cajero
1	✓ 1 cajero	Ejemplo1.p	0	1

Control seleccionado y valor que éste adopta para el primer escenario

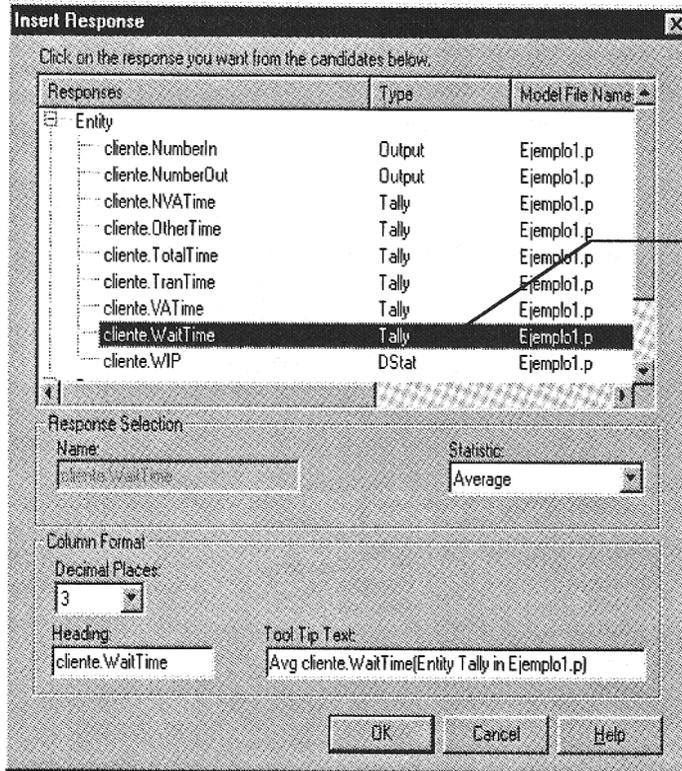
1.3 DEFINICIÓN DE VARIABLES DE RESPUESTA

Luego de seleccionar el control (factor controlable), la o las variables de respuesta (*Responses*) se incluyen en el proyecto mediante la ruta *Insert/Response*.



En el modelo existen cuatro tipos de categorías, de las cuales se pueden seleccionar las variables de respuesta, a saber: entidades, colas, recursos y sistema. El tiempo de espera, los tiempos en el sistema y el número de clientes en el sistema se toman de la categoría de entidades.

La primera respuesta que se incluye es el tiempo de espera de cada entidad llamada cliente. Este tiempo se encuentra consignado en el atributo *Nombre de la Entidad.WaitTime*, en el cual el nombre de la entidad es el nombre que se le da al atributo *Entity.Type*. En este caso, el tiempo de espera lo da el atributo *cliente.WaitTime*, ya que el *Entity.Type* recibe el nombre de cliente.



Tiempo total promedio de espera de un cliente en el sistema. Estadística elegida como variable de respuesta

Se puede seleccionar sólo una variable de respuesta a la vez. Cuando se presiona *ok* se vuelve a la ventana del proyecto, en la cual la hoja de datos de los escenarios se actualiza con la variable de respuesta adicionada:

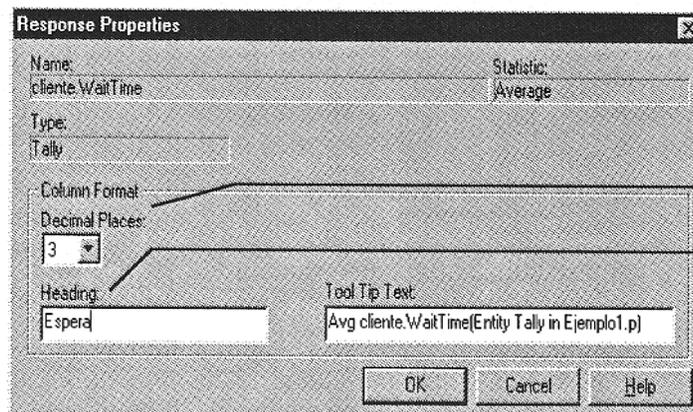
Scenario Properties				Control	
S	Name	Program File	Reps	cajero	cliente.WaitTime
1	✓ 1 cajero	1 Ejemplo1.p	0	1	...

Variable de respuesta seleccionada. No tiene ningún valor, ya que no se ha corrido ninguna simulación. El encabezado se puede cambiar en el campo *Heading* en el cuadro de adición de respuesta o al hacer clic derecho y editar sus propiedades

Si se hace clic derecho sobre la celda de la variable de respuesta, se pueden editar sus propiedades



Cuadro de diálogo de las propiedades de la variable de respuesta



Número de posiciones decimales

Encabezado

Scenario Properties				Control	
S	Name	Program File	Reps	cajero	Espera
1	✓ 1 cajero	1 Ejemplo1.p	0	1	---

Nuevo encabezado. Se presenta de manera más explícita para ofrecer información clara a quienes no están familiarizados con el modelo

Posteriormente se adicionan el tiempo en el sistema (*cliente.Totaltime*) y el número de clientes en el sistema (*cliente.WIP*).

Scenario Properties				Control	Responses		
S	Name	Program File	Reps	cajero	Espera	Tiempo en sistema	Cientes en sistema
1	✓ 1 cajero	1 Ejemplo1.p	0	1	---	---	---

Luego se adicionan las respuestas relacionadas con los recursos:

Número promedio de cajeros ocupados

De igual manera, la utilización del recurso completo se adiciona mediante la estadística *cajero.Utilization*.

1.4 CREACIÓN DE VARIOS ESCENARIOS A PARTIR DE UN ESCENARIO

La edición de la hoja de datos de escenarios ya finalizada se ve de la siguiente manera:

	Scenario Properties				Control	Responses				
	S	Name	Program File	Reps	cajero	Espera	Tiempo en sistema	Cientes en sistema	Numero ocupado	Utilizacion
1	<input checked="" type="checkbox"/>	1 cajero	1 Ejemplo1.p	0	1	---	---	---	---	---

Número de réplicas ejecutadas ← (Reps)
 Capacidad de atención del recurso cajero, MR (cajero) ← (cajero)
 cajero.Utilization → (Utilizacion)
 cajero.NumberBusy → (Numero ocupado)
 cliente.WaitTime → (Tiempo en sistema)
 cliente.TotalTime → (Espera)
 cliente.WIP → (Cientes en sistema)

Para crear diversos escenarios, se duplica el escenario creado y se cambia el valor del control que corresponda.

	Scenario Properties				Control	Responses				
	S	Name	Program File	Reps	cajero	Espera	Tiempo en sistema	Cientes en sistema	Numero ocupado	Utilizacion
1	<input checked="" type="checkbox"/>	1 cajero	1 Ejemplo1.p	0	1	---	---	---	---	---

Botón de selección de escenario → Cuando se hace clic derecho sobre el botón de selección de escenario, se ingresa a un menú contextual, el cual permite duplicar el escenario seleccionado

Go.
 Insert Scenario...
Duplicate Scenario(s)
 Delete Scenario(s)
 Scenario Properties...

Después de crear varios duplicados del primer escenario, éstos se editan según el nombre y el valor del control de cada uno.

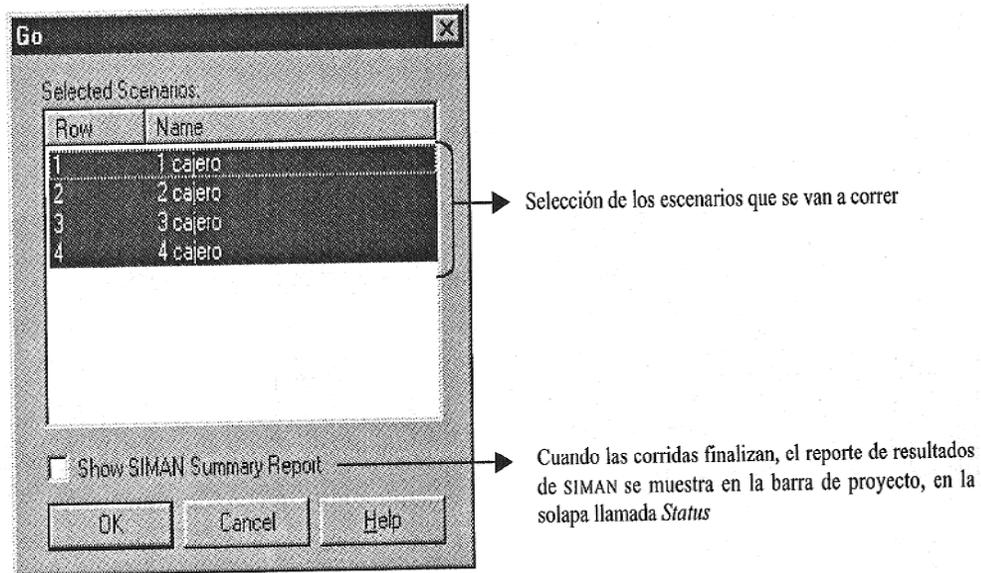
	Scenario Properties				Control	Responses				
	S	Name	Program File	Reps	cajero	Espera	Tiempo en sistema	Cientes en sistema	Numero ocupado	Utilizacion
1	<input checked="" type="checkbox"/>	1 cajero	1 Ejemplo1.p	0	1	---	---	---	---	---
2	<input checked="" type="checkbox"/>	2 cajero	1 Ejemplo1.p	0	2	---	---	---	---	---
3	<input checked="" type="checkbox"/>	3 cajero	1 Ejemplo1.p	0	3	---	---	---	---	---
4	<input checked="" type="checkbox"/>	4 cajero	1 Ejemplo1.p	0	4	---	---	---	---	---

Se crean diversos escenarios de acuerdo con la capacidad de atención del recurso cajero

1.5 CORRIDA DE LOS ESCENARIOS

Luego de definir los escenarios que se van a evaluar, se efectúan las corridas respectivas. Para ello se escoge el escenario que se desea estudiar y se presiona el botón de corrida  o clic derecho; luego se selecciona la opción *Go* a través del menú contextual o mediante la tecla F5.

Si se desean correr todos los escenarios, éstos se seleccionan y se presiona el botón de corridas o cualquiera de las acciones antes mencionadas para tal efecto. Cuando se hace esta operación aparece el cuadro de diálogo de corridas (*Go*).



Después que finalizan las corridas, la hoja de datos de los escenarios se actualiza con los resultados que para cada situación presentan las variables de respuesta, así como se muestra a continuación:

	Scenario Properties				Control	Responses				
	S	Name	Program File	Reps	cajero	Espera	Tiempo en sistema	Clientes en sistema	Numero ocupado	Utilización
1		1 cajero	1 Ejemplo1.p	5	1	7.044	11.659	2.036	0.782	0.782
2		2 cajero	1 Ejemplo1.p	5	2	0.083	4.778	0.818	0.804	0.402
3		3 cajero	1 Ejemplo1.p	5	3	0.000	4.695	0.804	0.804	0.288
4		4 cajero	1 Ejemplo1.p	5	4	0.000	4.695	0.804	0.804	0.201

Los valores que se muestran en cada escenario corresponden al promedio de los valores obtenidos en las réplicas.

En los resultados se aprecia que al contar con una capacidad de atención de 2, se reduce notablemente el tiempo en el sistema (resultado de la reducción en la espera). El número de cajeros ocupados en el escenario 1 indica que sólo un cajero puede estar capa-

citado para las condiciones del sistema, ya que es menor que la unidad en un grado significativo.

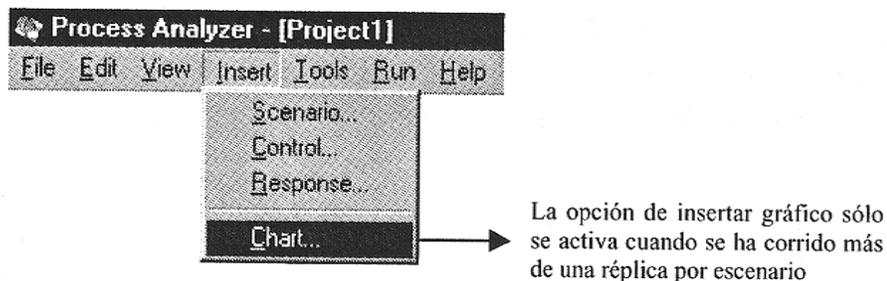
En el escenario 2 se aprecia que la inclusión de un cajero disminuye la utilización del recurso, ya que al incrementar la capacidad de atención del tiempo para las condiciones dadas, el tiempo ocioso se incrementa. Este hecho se refuerza por el número de recursos ocupados, que es de 0.804, lo cual indica que no se alcanza a ocupar uno completamente y que se cuenta con dos.

En los escenarios 3 y 4 los resultados son prácticamente los mismos, pues para las condiciones dadas un incremento en la capacidad no generará mejoras significativas, por el contrario, incrementará los costos de operación. En el caso de los recursos sólo se necesita 0.84 cajeros para cumplir, y se cuenta con 3 y 4. Resulta obvio que la utilización del servidor decrece a medida que se le incrementa la capacidad de atención (el exceso de capacidad con respecto a los niveles de servicio demandados disminuyen la utilización).

1.6 CREACIÓN DE GRÁFICOS

Se puede crear un gráfico que muestre el valor de una o más variables de respuesta, ya sea por réplica o por cada escenario. Esto se hace mediante la ruta *Insert/Chart*.

Importante. Sólo se generan gráficos cuando se tiene más de una réplica por escenario.



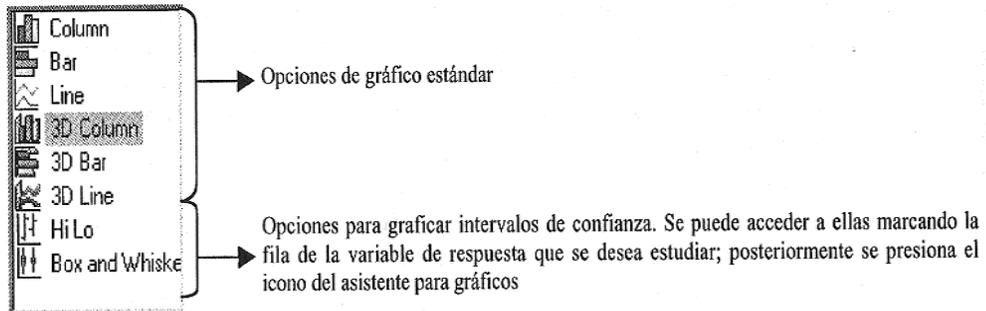
Los gráficos se pueden crear mediante el icono de acceso al asistente para gráficos:



El *Process Analyzer* cuenta con un asistente para la construcción de gráficos, el cual permite comparar resultados entre réplicas o a través de los escenarios, incluso seleccionando el mejor.

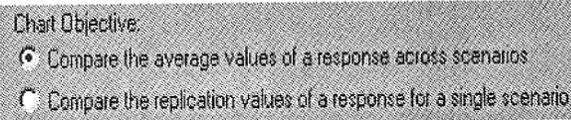
Si desde un gráfico de intervalos de confianza para una determinada variable de respuesta, y teniendo en cuenta todos los escenarios, se hace clic derecho en su encabezado, se selecciona la opción *Insert chart* en el menú contextual y, de esta manera, se ingresa al asistente para gráficos, entonces, aparecerán dos nuevas opciones de gráficos: intervalo de confianza y diagrama de caja.

Importante. Para graficar intervalos de confianza es preciso que para la variable de respuesta se hayan completado, por lo menos, tres réplicas por escenario.

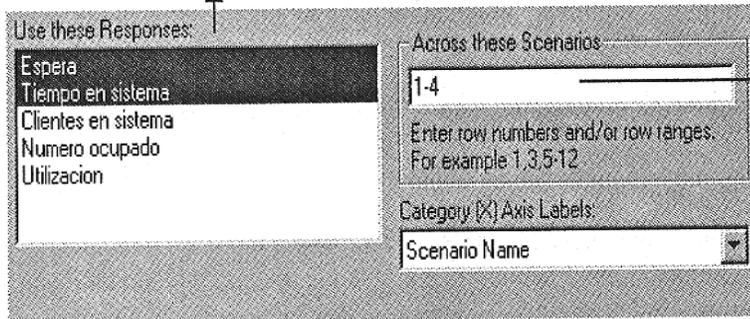


Suponga que se desea graficar el valor del tiempo de espera en el sistema a través de todos los escenarios:

- Inicialmente se hace clic en cualquier celda de las variables de respuesta, para habilitar el icono del asistente para gráficos.
- Luego se inicia el asistente mediante el icono .
- En el asistente se selecciona una variable de respuesta a través de varios escenarios; posteriormente se escoge el tipo de gráfico; en este caso se trata de uno de líneas. Luego se presiona el botón siguiente para continuar con el próximo paso.

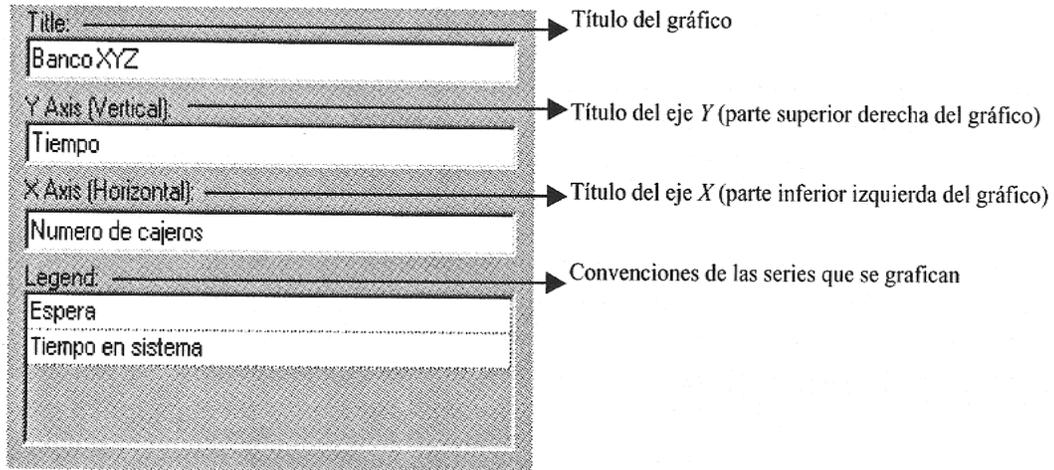


Se seleccionan las variables de respuesta que se desean graficar

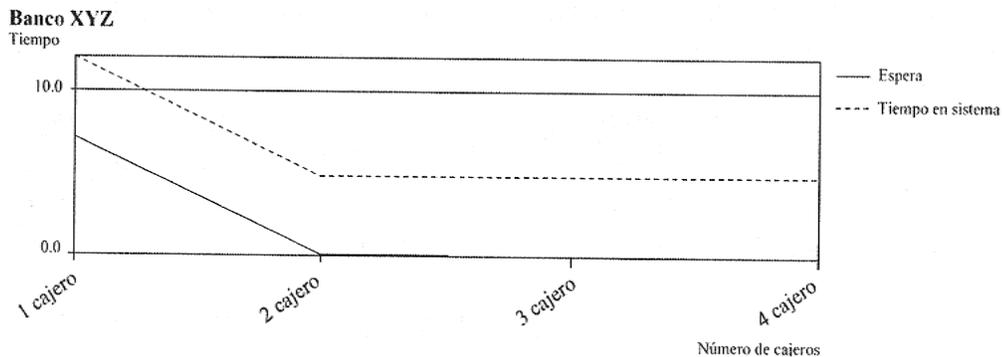


Se seleccionan los escenarios que se incluyen en el análisis. Los valores individuales son separados por comas y los rangos, por guión

- Se presiona el botón siguiente para pasar al próximo paso.
- El paso 3 en el asistente consiste en especificar los títulos para el gráfico.
- Se presiona el botón siguiente para iniciar el último paso del asistente.



- El último paso es útil cuando se grafica una sola variable de respuesta, ya que permite especificar al asistente que señale o resalte el mejor escenario según sea el caso (minimización o maximización de la variable de respuesta). En este caso no tiene mayor utilidad, así que se puede presionar el botón siguiente.
- Es posible cambiar opciones de formato del gráfico, como el espesor de línea o la alineación de los rótulos de las categorías del eje *X*.



El gráfico creado se muestra a continuación:

En el gráfico se aprecia que las mejoras notables se dan en el escenario 2, una mejora poco significativa se muestra en el escenario 3 y ninguna mejoría se presenta en el escenario 4.

La línea sólida, que corresponde a la espera, llega prácticamente a 0 en el escenario 2; por ello el tiempo en el sistema (línea punteada) tiende a ser el promedio del tiempo de atención (4.69), el cual proviene de una distribución exponencial con media de 4.5.

1.7 UTILIZACIÓN DE UNA VARIABLE COMO RESPUESTA

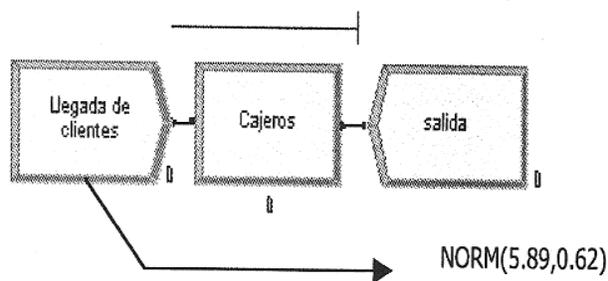
Hasta ahora se ha encontrado que con dos cajeros el tiempo de espera de los clientes corresponde casi a 0, por lo tanto, para un banco sería conveniente decidirse por esta opción. Otro análisis que podría realizarse es comprobar los efectos de cambios en factores externos (en la realidad incontrolable) para las condiciones actuales o propuestas.

Suponga que en el caso en estudio se ha fijado un tiempo de espera medio de 3 minutos como estado de atención al cliente. En las condiciones propuestas, ¿qué pasaría si la llegada de clientes se incrementa? El incremento se asumirá como una disminución en la media de llegada de una unidad; es decir, para la alternativa propuesta de dos cajeros se tienen los siguientes escenarios:

Escenario	Valor de la media de llegadas
Escenario 1	5.89
Escenario 2	4.89
Escenario 3	3.89
Escenario 4	2.89
Escenario 5	1.89

Para resolver este caso se debe modificar el modelo mediante el *Process Analyzer* de tal manera que la media de la distribución empleada para generar las llegadas sea una variable.

La expresión que se utiliza en el módulo *Create* para generar las llegadas con el modelo original es la siguiente:



Para el análisis, esa expresión se debe cambiar por la siguiente: $NORM(media, 0.65)$, en la que *media* es una variable cuyo valor inicial es 5.89, el cual permanecerá constante a través de toda la simulación pues no existe lógica que incida sobre esta variable.

La variable *media* se debe definir en la hoja de datos de variables del panel de procesos básicos:



Variable

Cuando se hace doble clic en el lugar indicado se adiciona una fila para la variable *media*, así como se presenta a continuación:

	Name	Rows	Columns	Clear Option	Initial Values	Report Statistics
1	media			System	1 rows	<input type="checkbox"/>

Se especifica el valor inicial que tendrá la variable *media* (valor desde el tiempo o de simulación)

	Initial Value
1	5.89

Double-click here to add a new row

El modelo con las respectivas modificaciones se guarda con otro nombre; en este caso en particular se trata del ejemplo 9.1. Debido a las modificaciones, es necesario que se realice una revisión del modelo con el fin de generar el archivo actualizado para emplearlo en el *Process Analyzer*.

En el *Process Analyzer* se preparan los escenarios de manera análoga como se hizo en el estudio que se realizó primero. En el caso de este ejemplo se debe adicionar un control más, el cual corresponde a la variable que representa el tiempo entre llegadas:

Las variables entran en la categoría *especificadas por el usuario*

Tipo de dato. Puede ser entero o real. En este caso se elige real, ya que la variable puede adoptar valores decimales

Se selecciona el número de posiciones decimales

Encabezado de la columna para el control especificado

A continuación se muestra la hoja de datos de los escenarios que se elaboró para el análisis de la incidencia de la llegada para la política propuesta.

	Scenario Properties				Controls		Responses				
	S	Name	Program File	Reps	cajero	media	Espera	Tiempo en sistema	Clientes en sistema	Numero ocupado	Utilizacion
1	✓	Escenario1	1: Ejemplo1_	0	2	5.8900	---	---	---	---	---
2	✓	Escenario2	1: Ejemplo1_	0	2	4.8900	---	---	---	---	---
3	✓	Escenario3	1: Ejemplo1_	0	2	3.8900	---	---	---	---	---
4	✓	Escenario4	1: Ejemplo1_	0	2	2.8900	---	---	---	---	---
5	✓	Escenario5	1: Ejemplo1_	0	2	1.8900	---	---	---	---	---

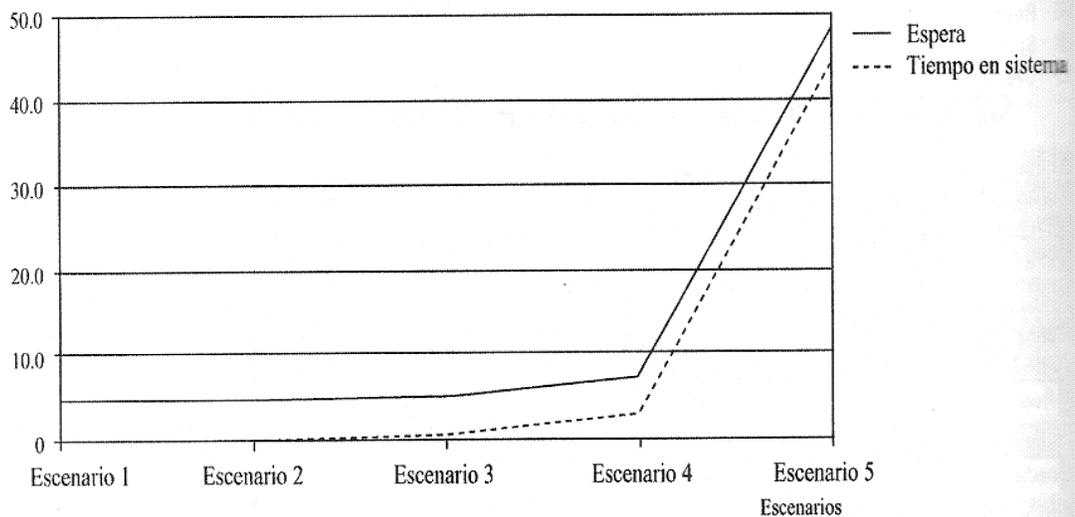
Escenarios propuestos para probar la robustez de la decisión ante cambios en factores incontrolables en la realidad

Cuando se corren los diferentes escenarios, se tienen los siguientes resultados:

	Scenario Properties				Controls		Responses				
	S	Name	Program File	Reps	cajero	media	Espera	Tiempo en sistema	Clientes en sistema	Numero ocupado	Utilizacion
1	✗	Escenario1	1: Ejemplo1_	5	2	5.8900	0.083	4.776	0.816	0.604	0.402
2	✗	Escenario2	1: Ejemplo1_	5	2	4.8900	0.216	4.865	1.002	0.958	0.479
3	✗	Escenario3	1: Ejemplo1_	5	2	3.8900	0.439	5.001	1.299	1.187	0.593
4	✗	Escenario4	1: Ejemplo1_	5	2	2.8900	2.982	7.506	2.609	1.577	0.789
5	✗	Escenario5	1: Ejemplo1_	5	2	1.8900	44.130	48.591	25.555	1.988	0.994

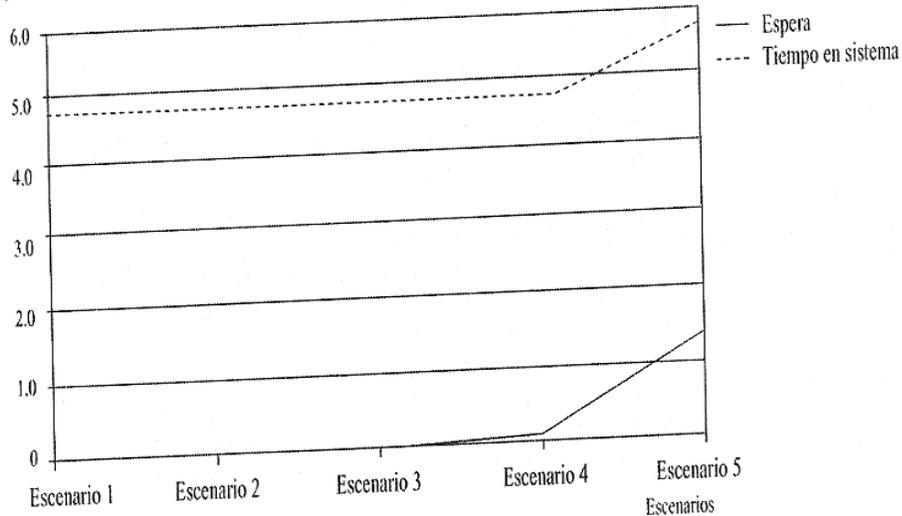
Como se aprecia en los resultados, esta política sólo soporta hasta una variación de 3 minutos en la media del tiempo entre llegadas de clientes; además, cumple con los estándares de atención previamente establecidos. A continuación se presenta una representación gráfica de esta situación:

Banco XYZ
Tiempos



Un análisis similar se podría hacer en el caso de la implementación de 3 cajeros; ante los mismos cambios en la llegada de clientes, esta práctica será más sensible pero más costosa. A continuación se presenta un resumen gráfico de esta situación:

Banco XYZ
Tiempos



Como se alcanza a apreciar en la anterior gráfica, la situación más apremiante (media = 1.89 minutos) ocasiona un tiempo de espera de menos de 2 minutos.

En la realidad existen muchos más factores que controlar y más situaciones o escenarios que contemplar; por ello *Process Analyzer* se convierte en una herramienta de gran utilidad.

2. RECOLECCIÓN DE ESTADÍSTICAS

Existen situaciones en las cuales el interés del análisis se puede centrar en una parte del modelo, bien sea para estudio o para validación. Es posible que se deba realizar un conteo de entidades en determinada parte del modelo, medir el tiempo medio de llegadas de entidades a una estación o proceso en un punto medio del modelo, o medir el tiempo que se toman las entidades en ir de un punto del modelo a otro. Estas estadísticas se recolectan y muestran en los reportes bajo la categoría de estadísticas que define el usuario (*User Defined*).

Para recolectar estadísticas en cualquier parte del modelo se emplea el módulo *Record*.



Record

Campo	Descripción
<i>Name</i>	Nombre del módulo. Debe ser único y preferiblemente decir algo sobre la función que desempeña.
<i>Type</i>	Tipo de estadística que se recolectará.
<i>Count</i>	Esta función permite que el módulo realice un conteo de entidades a medida que pasa a través de él.
<i>Value (Count)</i>	Cuando se emplea la función <i>Count</i> se activa el campo <i>Value</i> , el cual registra la cantidad en la que se incrementa el contador por entidad contada, es decir, de 1 en 1 o de 2, 5 en 5, etc.
<i>Counter name</i>	Nombre que se le da al contador para que se le identifique en los reportes. Si no se especifica, adopta el nombre del módulo.
<i>Entity Statistics</i>	Recolecta de estadísticas de la entidad hasta la instancia del modelo en que éstas se colocan y se muestran en los reportes. Si varios módulos <i>Record</i> tienen este mismo tipo de acción, sólo la información recolectada en cada uno de ellos se mezcla y aparece en los reportes.
<i>Time intervalo</i>	Recolecta de estadísticas relacionadas con un intervalo de tiempo entre dos puntos del modelo. La entidad debe contener un atributo que incluya el tiempo en el que pasó por el primer punto, para que en este módulo se evalúe en comparación con el tiempo actual (<i>Now</i>) y así se determine el intervalo de tiempo transcurrido.
<i>Attribute name</i>	Nombre del atributo que contiene el tiempo en el que la entidad pasó por el punto inicial del intervalo.
<i>Tally name (Time interval)</i>	Nombre de la estadística en la cual se recolectan los datos obtenidos. Si no se coloca nombre, el programa asume el del módulo.
<i>Time between</i>	Recolecta de tiempo entre llegadas de entidades al módulo. La primera entidad que llega no se tiene en cuenta en el cálculo, ya que ésta inicia el conteo ($n-1$ datos de tiempo entre llegadas).
<i>Tally name (time between)</i>	Nombre de la estadística en la cual se recolectan los datos obtenidos. Si no se coloca nombre, el programa asume el del módulo.
<i>Expression</i>	Calcula una expresión para que sea reportada como estadística. Una expresión puede ser, según las necesidades del analista, el valor de un atributo multiplicado por otro o por una constante, el mínimo valor entre varios atributos de la entidad, etc.
<i>Tally name (Expression)</i>	Nombre de la estadística en la cual se recolectan los datos obtenidos. Si no se coloca nombre, el programa asume el del módulo

Las estadísticas observables o *Tallies* arrojan un promedio aritmético normal de una variable, es decir, la suma del total de datos dividido entre el número de observaciones.

Las estadísticas de variables arrojan un promedio ponderado con base en el tiempo, es decir, la suma de los valores que adopta la variable multiplicado por el tiempo en que ésta estuvo en ese valor (peso) y dividido por el tiempo total.

Las estadísticas de la entidad (*Entity Statistics*) corresponden a todos los tiempos según su categoría (tiempo total, valor agregado, espera, etc.). Las estadísticas de utilización de recursos, transportes, colas, entre otras, se recolectan normalmente porque son del tipo global.

Ejemplo 9.3

■ Descripción

Considere un sistema en el que se cuenta con 4 procesos en serie: A, B, C y D, los cuales requieren de los recursos A, B, C y D, respectivamente. Los tiempos de procesamiento en minutos se muestran a continuación:

Proceso/Recurso	Tiempo
A	Normal (8,0.89)
B	Uniforme (4,12)
C	Triangular (5,10,15)
D	Exponencial (5)

El tiempo entre llegadas de entidades al sistema en referencia sigue una distribución uniforme entre 1 y 20 minutos. El sistema será simulado en 2.400 minutos. Las unidades de tiempo base serán minutos.

■ Desarrollo

La construcción de este modelo es bastante sencilla; por ello sólo se muestra una tabla resumen con los datos empleados más relevantes:

Módulo Create

Campo	Valor
<i>Name</i>	Entrada
<i>Entity type</i>	Parte
<i>Type</i>	<i>Expression</i>
<i>Expression</i>	UNIF (1,20)
<i>Time units</i>	Minutos
El resto de campos se deja con los valores por omisión (<i>Default</i>)	

Módulos Process

Proceso A	
Campo	Valor
<i>Name</i>	Proceso A
<i>Action</i>	<i>Seize, Delay, Release</i>
<i>Resource name</i>	Recurso A
<i>DelayType</i>	Normal
<i>Units</i>	minutos
<i>Value (mean)</i>	8
<i>Std Dev</i>	0.89
El resto de campos se deja con los valores por omisión (<i>Default</i>).	

Proceso B	
Campo	Valor
<i>Name</i>	Proceso B
<i>Action</i>	<i>Seize, Delay, Release</i>
<i>Resource name</i>	Recurso B
<i>DelayType</i>	<i>Uniform</i>
<i>Units</i>	minutos
<i>Minimum</i>	4
<i>Maximum</i>	8
El resto de campos se deja con los valores por omisión (<i>Default</i>).	

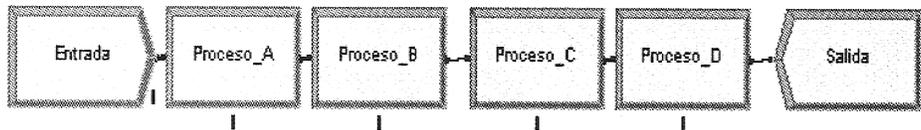
Proceso C	
Campo	Valor
Name	Proceso C
Action	Seize, Delay, Release
Resource name	Recurso C
DelayType	Triangular
Units	minutos
Minimum	5
Value (Most likely)	10
Maximum	15
El resto de campos se deja con los valores por omisión (Default).	

Proceso D	
Campo	Valor
Name	Proceso D
Action	Seize, Delay, Release
Resource name	Recurso D
DelayType	Expression
Units	minutos
Expression	EXPO (5)
El resto de campos se deja con los valores por omisión (Default).	

Módulo Dispose

Campo	Valor
Name	Salida
Record Entity Statistics	Casilla de verificación seleccionada

A continuación se muestra la secuencia de módulos en ARENA®:



a. Conteo de entidades

Suponga que se desea saber el número de entidades que llegaron al proceso D. Para ello se agrega un módulo *Record* antes del proceso D y se configura como tipo *Count*, así como se muestra a continuación:

Record

Record

Name:

Type:

Value: Record into Set

Counter Name:

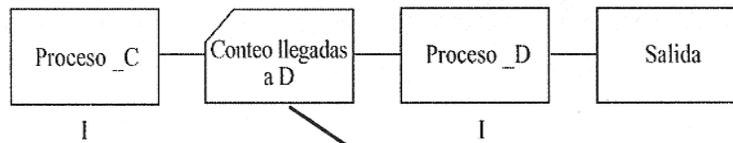
Nombre del módulo. Debe especificar la función que cumple en el modelo

Tipo de módulo *Record*. En este caso se configura como contador (*Count*)

Incremento del contador. En este caso se desea contar de 1 en 1

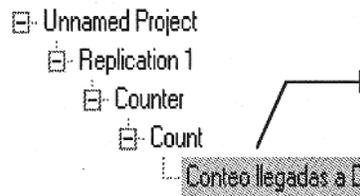
Nombre del contador. Este nombre identificará al contador en los reportes

A continuación se muestra la secuencia de módulos lógicos en ARANA®:



Módulo *Record* que se agrega para contar las entidades que llegan al proceso D

El modelo se corre en el tiempo especificado. A continuación se muestran los resultados:



De la barra exploradora de reportes se despliega el informe de interés de la categoría contadores (*Count*). Como se observa, está identificado con el nombre que se le dio en el módulo *Record*

De la barra de proyectos (*Reports*) se selecciona la categoría de estadísticas que define el usuario

En la ventana de informe se muestra el valor final del contador *Conteo llegadas a D*:

Replication 1	Start Time:	0,00	Stop Time:	2.400,00
----------------------	-------------	------	------------	----------

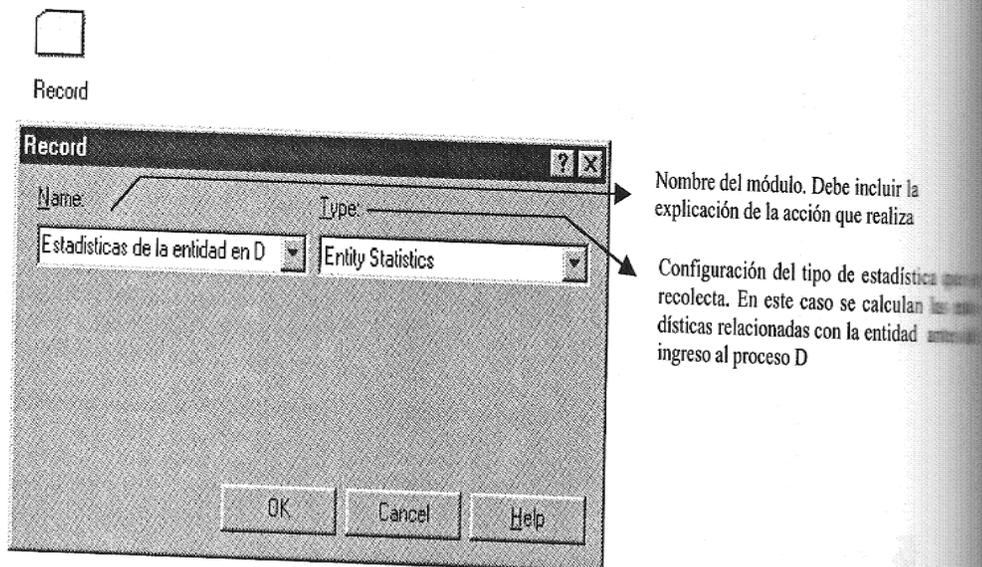
Counter

Count	Value
Conteo llegadas a D	228.00

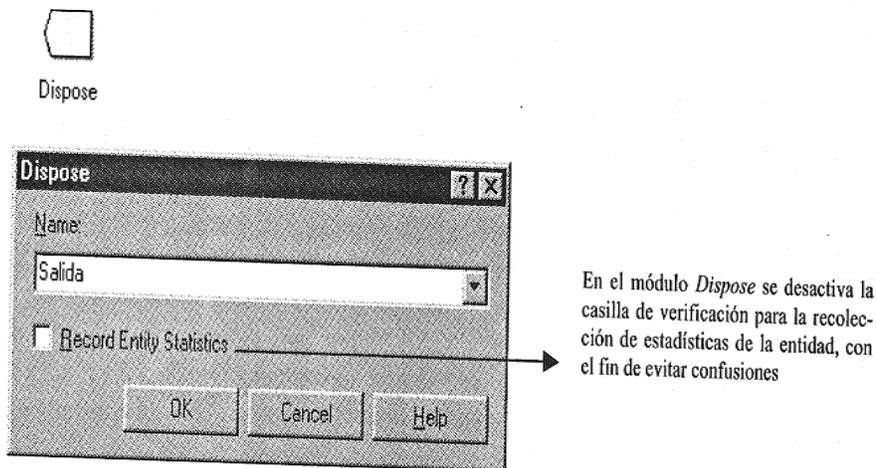
Valor final del Contador de llegadas a D. Es decir, en este caso llegaron 228 entidades al proceso D

b. Recolección de estadísticas de la entidad

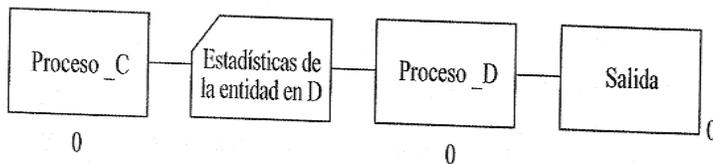
Suponga que se desea recolectar estadísticas de la entidad sin tener en cuenta el proceso en el recurso D. Para ello se agrega un módulo *Record* antes del módulo de proceso D y se configura para recolectar estadísticas de la entidad (*Entity Statistics*). A continuación se muestra esta operación:



Cuando se emplea este tipo de estadísticas, se debe cuidar que la casilla de verificación para recolección de estadísticas de la entidad quede en blanco en los módulos *Dispose* utilizados en el modelo. Si no se hace de esta manera, se estarán mezclando las dos estadísticas. Es decir, cada observación se debe almacenar y contabilizar parcialmente por el módulo *Record* (recolecta y procesa información hasta ese punto), y al final se debe almacenar y contabilizar nuevamente pero de forma total incurriendo en errores.



La secuencia de módulos lógicos en ARENA[®] se muestra a continuación:



El modelo se corre en el tiempo especificado. A continuación se muestran los resultados:

Replication 1 Start Time: 0,00 Stop Time: 2.400,00 Time Units: Minutes

Entity Detail Summary

Time

	VA Time	NVA Time	Transfer Time	Wait Time	Other Time	Total Time
parte	25.73	0.00	0.00	23.00	0.00	48.72
Total	25.73	0.00	0.00	23.00	0.00	48.72

Tiempo de espera para las entidades sin incluir el proceso D

Tiempo total para las entidades sin incluir el proceso D. En otras palabras, sería como si en D estuviera la salida

Estos datos se pueden comparar con los del modelo original; de esta manera, y ante las modificaciones realizadas, se podrán verificar las diferencias en los resultados.

Replication 1 Start Time: 0,00 Stop Time: 2.400,00 Time Units: Minutes

Entity Detail Summary

Time

	VA Time	NVA Time	Transfer Time	Wait Time	Other Time	Total Time
parte	31.07	0.00	0.00	24.42	0.00	55.49
Total	31.07	0.00	0.00	24.42	0.00	55.49

Tiempo de espera para las entidades incluyendo el proceso D

Tiempo total para las entidades incluyendo el proceso D. Las estadísticas se recolectan en la salida del sistema (después del proceso D)

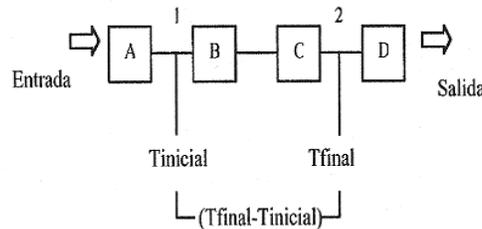
c. Intervalo de tiempo entre dos puntos del modelo

Suponga que se tiene interés en medir el tiempo que demora una entidad en ir desde la salida del proceso A hasta la salida del proceso C. Para ello se debe seguir el siguiente procedimiento:

- Tomar el tiempo (tiempo simulado) en el que una pieza sale del proceso A, *Tinicial*.

- Tomar el tiempo (tiempo simulado) en el que la misma pieza sale del proceso B, T_{final} .
- La longitud del intervalo está dada por la diferencia $T_{final}-T_{inicial}$ para cada entidad.
- La longitud del intervalo se acumula y se divide entre el total de entidades observadas; luego se tiene el promedio del tiempo entre la salida de A y la salida de C (estadística observable o *Tally*).

El anterior procedimiento se representa gráficamente de la siguiente manera:



Se debe crear un atributo en el cual se almacenará el tiempo en el que la entidad pasa por el punto 1 de la salida del proceso A. En este caso en particular, este atributo recibe en ese instante el nombre de $t_{inicial}$ y el valor del tiempo de simulación ($tnow$). A continuación se muestra esta operación:

La imagen muestra un cuadro de diálogo con el título "Assignments". En la parte superior izquierda hay un icono de un cuadro vacío etiquetado como "Assign". El cuadro de diálogo tiene los siguientes campos:

- Type:** Un menú desplegable con "Attribute" seleccionado.
- Attribute Name:** Un menú desplegable con "t_inicial" seleccionado.
- New Value:** Un campo de texto con "tnow" escrito.

En la parte inferior hay tres botones: "OK", "Cancel" y "Help".

Las anotaciones son:

- Una flecha apunta desde el icono "Assign" a un texto: "Tipo de asignación. En este caso se desea crear un atributo, ya que el tiempo en que una entidad pasa por el punto 1 es propio de cada entidad".
- Una flecha apunta desde el campo "Attribute Name" a un texto: "Variable que representa el tiempo actual de simulación".
- Una flecha apunta desde el campo "New Value" a un texto: "Esta variable se puede obtener así: se hace clic derecho en el cuadro de texto *New value*, luego se ingresa al asistente para construcción de expresiones (*Expression builder*) y se busca la variable mediante la siguiente ruta: *Simulation control variables/Current simulation time*".
- Una flecha apunta desde el campo "Attribute Name" a un texto: "Nombre del atributo. Todas las entidades que pasen por este módulo tendrán un atributo llamado $t_{inicial}$, pero éste tendrá un valor diferente en cada una de ellas de acuerdo con el tiempo en que pasaron por el punto 1".

Para evaluar el intervalo se adiciona un módulo *Record* en el punto 2 y su acción se configura como tipo *Time intervalo*. Esta acción permite evaluar un intervalo de tiempo teniendo en cuenta el tiempo actual y un tiempo anterior previamente salvado en un atributo ($tnow-t_{inicial}$). También se crea una estadística observable o *Tally*, en la cual se almacenarán los intervalos de tiempo de las diferentes entidades, con el fin de que se

promedien y se muestren en los reportes. La edición de este módulo se muestra a continuación:



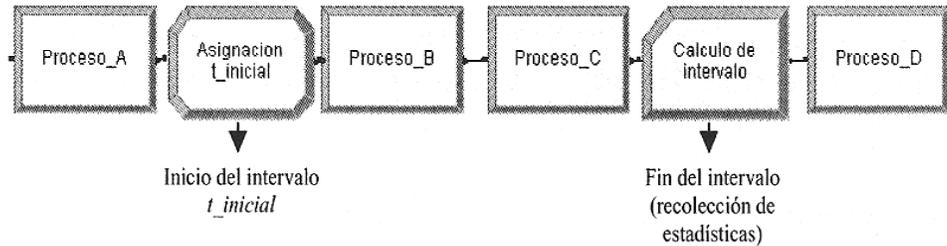
Record

Tipo de módulo *Record*. En este caso en particular se emplea para medir un intervalo de tiempo

Nombre del atributo que contiene el tiempo inicial del intervalo

Nombre de la estadística. Toma automáticamente el nombre del módulo

A continuación se muestra la secuencia de módulos lógicos en ARENA®:



El modelo se corre en el tiempo especificado. A continuación se muestran los resultados (*User defined*):

Replication 1	Start Time:	0,00	Stop Time:	2.400,00	Time Units:	Minutes
----------------------	-------------	------	------------	----------	-------------	---------

Tally

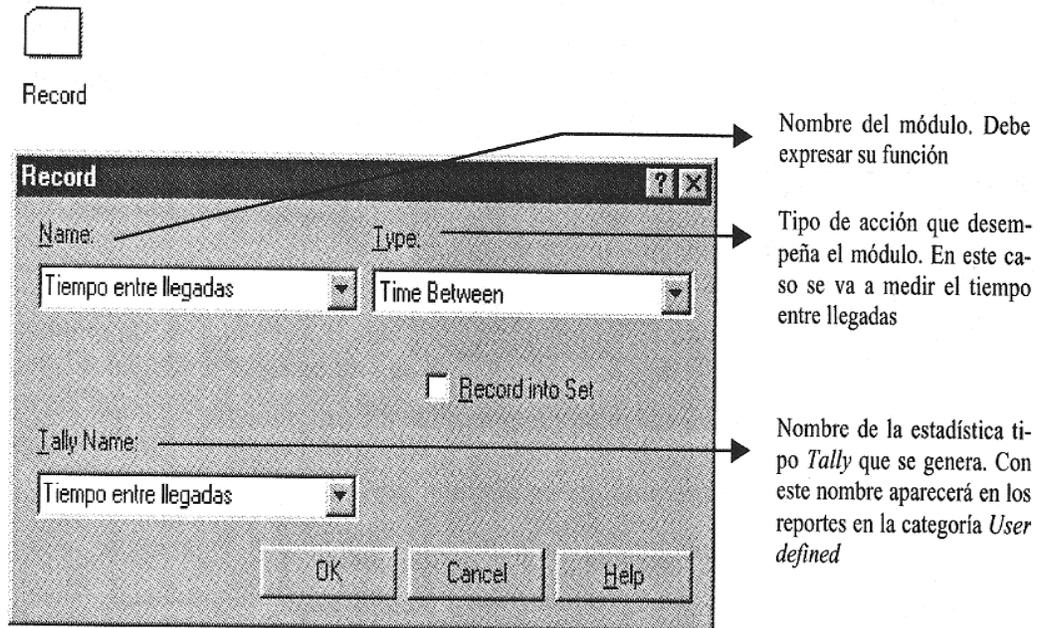
Interval	Average	Half Width	Minimum	Maximum
Calculo de intervalo	37.3059	(Insufficient)	13.2232	74.8306

Tiempo promedio en el que demora una entidad en ir desde la salida de A hasta la salida de C

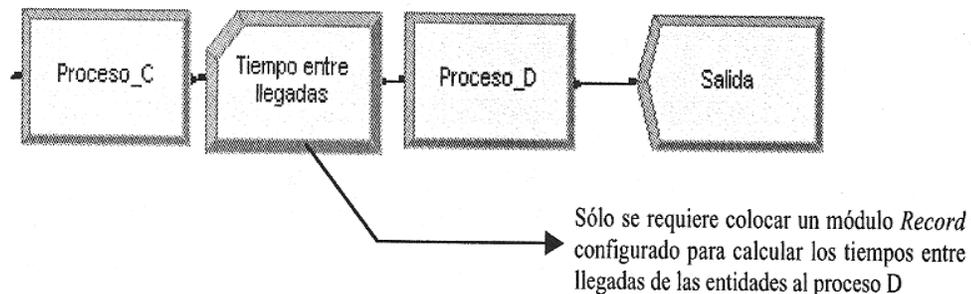
El tamaño de la muestra recogida con respecto a la variabilidad de los datos no es suficiente para calcular un intervalo de confianza

d. Tiempo entre llegadas

Suponga que se desea conocer la tasa de llegadas de entidades al proceso D. Se debe, entonces, agregar un módulo *Record* antes del proceso D y configurar su acción para medir el tiempo entre llegadas (*Time between*). Los $n-1$ datos (tiempo entre llegadas de n entidades) se almacenan en una estadística tipo *Tally* para su procesamiento y reporte. La edición del módulo *Record* se muestra a continuación:



A continuación se muestra la secuencia de módulos lógicos en ARENA®:



El modelo se corre en el tiempo especificado. A continuación se muestran los resultados bajo la categoría *User defined* en los reportes *Category overview* (en una sola réplica arroja los mismos resultados):

Between	Average	Half Width	Minimum Value	Maximum Value
Tiempo entre llegadas	10.4502	(Insuficient)	5.4036	24.0650

Promedio del tiempo entre llegadas de entidades al proceso D

Así como se muestra en el apartado *intervalo de tiempo entre dos puntos del modelo* (p.161), el tamaño de la muestra recogida con respecto a la variabilidad de los datos no es suficiente para calcular un intervalo de confianza

El modelo se podría correr en un tiempo arbitrario pero mayor que el actual; por ejemplo, 10.000 minutos; de esta manera se obtendría un tamaño de muestra que permitiría calcular un intervalo de confianza:

Between	Average	Half Width	Minimum Value	Maximum Value
Tiempo entre llegadas	10.6768	0,225959321	5.4036	27.9381

Promedio del tiempo entre llegadas al proceso D

Mitad del intervalo de confianza con $\alpha=0.05$

La función *Expression* permite almacenar en una estadística observable (*Tally*) el valor que tiene una determinada expresión cuando una entidad pasa por el módulo. Se debe tener en cuenta que, al final, el valor promedio resulta de la suma del valor de cada observación más el número de entidades (observaciones) que se tienen.

Una expresión puede ser una relación matemática entre valores de atributos de una entidad o de variables del modelo. Por ejemplo, si las entidades, en cierto punto, representan una lámina en forma triangular y poseen dos atributos llamados *base* y *altura*, la expresión que se va a calcular podría ser el área $base*altura/2$.

La recolección de estadísticas mediante el módulo *Record* cobra importancia cuando se desean analizar datos con las herramientas que contiene el ambiente de trabajo ARENA®, como el *Process Analyzer* o el *Output Analyzer*.

Adicionalmente, el programa arroja la tabla resumen del análisis de varianza, así como se muestra a continuación:

One-Way ANOVA Table: ANOVA Método proceso C

- L1 - Cálculo de intervalo
- L2 - Cálculo de intervalo
- L3 - Cálculo de intervalo

SOURCE OF VARIATION	SUM SQUARES	DF	MEAN SQUARES	F-EXP
BETWEEN TREATMENTS	4.899e+004	2	2.45e+004	39.129
ERROR (W.TREATMENTS)	4.301e+005	687	626	
TOTAL	4.791e+005	689	F-CRIT	3.008

REJECT H0 => NOT ALL MEANS ARE EQUAL AT 0.05 LEVEL

En la tabla de ANOVA se puede apreciar que la varianza entre tratamientos es diferente a la varianza dentro de los tratamientos, por ello se puede asumir que no todas las medias son iguales (por lo menos una no es igual).

Otro tipo de gráfica que es un poco más sencilla que las anteriores es la de tipo *Plot*, la cual representa la variable de respuesta escogida (valores individuales) contra el tiempo simulado.



Plot

Existe otra función que es de utilidad en los casos en los que se requiera formar subgrupos o truncar archivos para limitar el número de observaciones a un tamaño específico, o tener en cuenta solamente las observaciones durante cierto intervalo de tiempo. Esta función es *Batch/Truncate*.



Batch/Truncate

Existe también la opción de ver los datos que se recolectaron mediante el menú *Graph/Table*. Esta función permite que aparezcan en la pantalla dos columnas; la primera corresponde al valor de la conservación y la segunda, al tiempo en el que se registró la observación.

Los archivos **.dat* no son manejados por otros programas; también se pueden exportar para que puedan ser procesados en otras aplicaciones mediante el menú *File/Data File/Export*.

3. OUTPUT ANALYZER/ANALIZADOR DE DATOS DE SALIDA

La simulación genera una historia artificial de un sistema y trata de aproximar el comportamiento de la realidad por medio de números aleatorios para la generación de las variables de entrada, ejemplo, tiempos entre llegadas, tiempos de procesos, etc. Por esta razón, las variables de respuesta o datos de salida se comportan de manera aleatoria, es decir, exhiben cierta variabilidad que se debe contemplar en el momento de interpretar los resultados que arroja un modelo. De esta manera, se crea la necesidad de realizar un análisis estadístico de los datos de salida mediante histogramas, gráficos de promedio acumulado, intervalos de confianza, correlogramas, entre otros. También, cuando se comparan dos diseños o situaciones en términos de una variable de respuesta, se recurre a pruebas de diferencia de medias y a análisis de varianza (un factor). Este tipo de análisis se puede realizar de manera rápida y fácil mediante el *Output Analyzer* (Analizador de datos de salida), que viene incluido en el ambiente ARENA®.

Para desarrollar análisis con esta herramienta se hace necesario tener un archivo con los datos de salida del modelo; archivos como éste se generan en una o más corridas del modelo. Los archivos de salida de datos se crean cuando al programa se le especifican el nombre y la extensión del archivo en el cual se quieren guardar las observaciones individuales de una estadística en particular; por ejemplo, si se quisiera tomar el tiempo de flujo (tiempo en el sistema) de una entidad como medida del desempeño de un sistema, entonces, las entidades se almacenan en un archivo de salida, el cual contendría para cada entidad datos grabados de tiempo de flujo; se pueden graficar estos datos, hallar el promedio acumulado, construir un histograma, etc.

El *Output Analyzer* no se encuentra en la instalación estándar del programa. Para contar con esta herramienta se debe seleccionar la opción de instalación personalizada y activar la casilla de verificación correspondiente.

3.1 GENERACIÓN DE ARCHIVOS

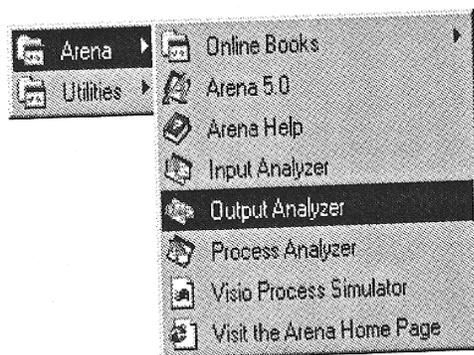
Los archivos que utiliza el analizador de datos de salida deben tener un formato especial y una extensión *.dat* para que puedan ser reconocidos. Estos archivos se crean mediante el módulo de datos *Statistic* del panel de procesos avanzados, y se van llenando de información a medida que avanza la simulación y las réplicas que se hagan de la corrida.



Statistic

Campo	Descripción
<i>Name</i>	Nombre de la estadística
<i>Type</i>	Tipo de estadística. Puede ser observable (<i>Tally</i>), frecuencia de ocurrencias (<i>Frecuencias</i>), etc.
<i>Expression/name</i>	Expresión o nombre de la estadística que se genera. Es posible que aparezcan otros campos adicionales según el tipo de información que se recolecta.
<i>Report label</i>	Rótulo que aparece en los reportes para identificar la estadística. Toma automáticamente el valor del campo <i>Name</i> , pero puede ser modificable.
<i>Output file</i>	Nombre del archivo de salida en el que se aclararán los datos para su análisis. Este campo se debe llenar con el nombre, seguido de la extensión <i>.dat</i> ; ejemplo, <i>Info.dat</i> .

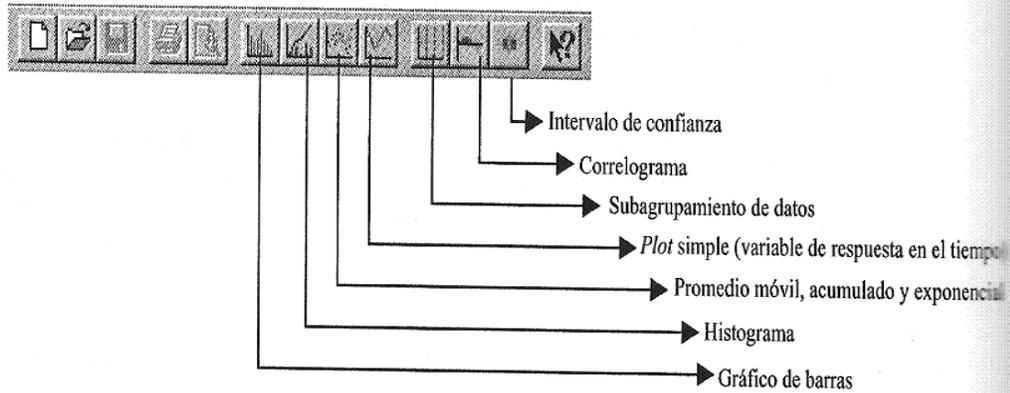
Se ingresa al analizador de datos de entrada a través del menú de ARENA[®] que se crea por defecto en la instalación del programa:



Ingreso al analizador de datos de salida. La aplicación se encuentra en el mismo directorio del programa *Arena.exe*

Output Analyzer

Una vez se está dentro del *Output Analyzer*, se encuentra la barra estándar, la cual contiene casi todas las instrucciones de los análisis que se pueden realizar con esta herramienta:



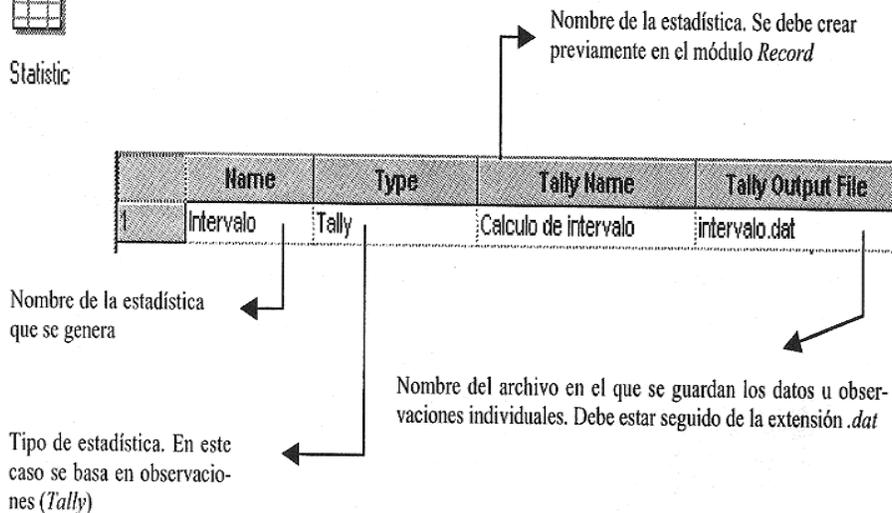
3.2 GENERACIÓN DEL ARCHIVO *.DAT

En el módulo *Record* se ha creado una estadística observable o *Tally*, que se llama tiempo entre llegadas, la cual contiene el total de datos recolectados. Estos se almacenarán en un archivo para su posterior análisis.

En el módulo de datos *Statistics* del panel de procesos avanzados se agrega una nueva estadística (agregando una fila), así como se muestra a continuación



Statistic



El modelo se ejecuta por el tiempo especificado (2.400 minutos) para llenar el archivo con los datos de la corrida.

3.3 GENERACIÓN DE HISTOGRAMA



Suponga que se desea construir un histograma con los datos recolectados.

El reporte de entidades muestra que entraron 233 (*Number in*) y salieron 227 (*Number out*). Se puede decir que por lo menos la estadística contiene un total de 227 observaciones correspondientes a las entidades que fueron retiradas del sistema en el momento de terminar la corrida; por ello, un histograma que se construya con estos datos tendrá \sqrt{n} datos, que en este caso es 15 intervalos.

Del informe de estadísticas que define el usuario se determina el ancho del intervalo del histograma, dividiendo el rango de las observaciones (max-min) entre el número de intervalos o clases del histograma, que en este caso es 15.

Interval	Average	Half Width	Minimum	Maximum
Calculo de intervalo	37.3059	(Insufficient)	13.2232	74.8306

Se construirá un histograma de 15 clases; el ancho de cada clase será 4.16.

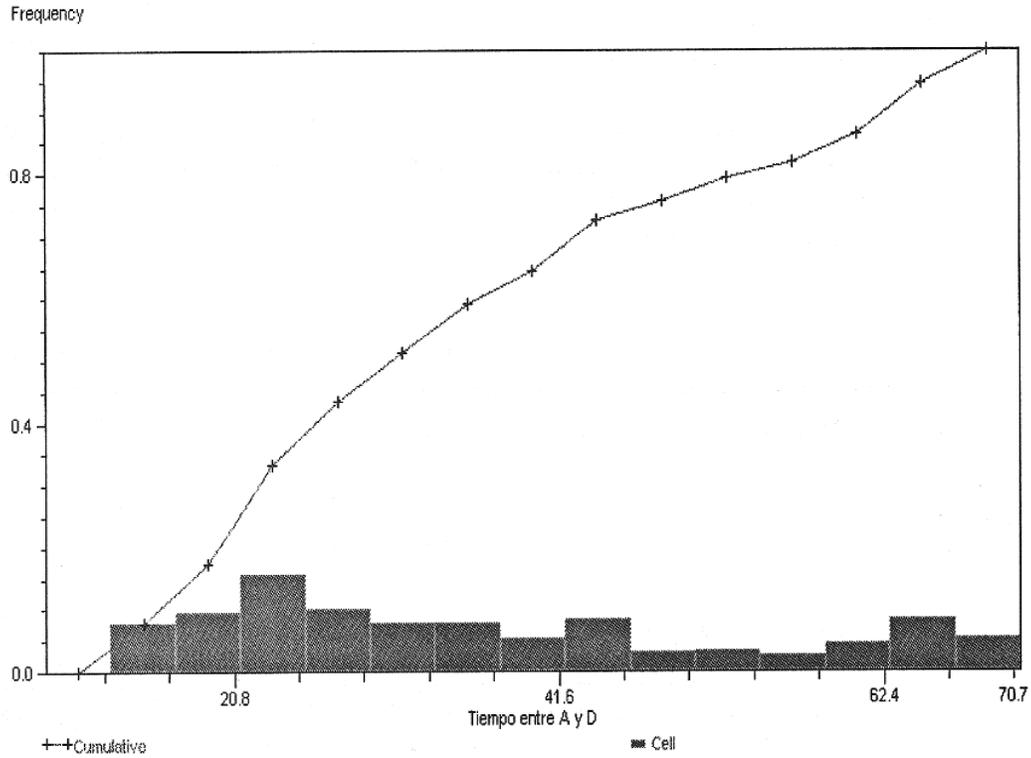
Importante. Cuando se construyen histogramas por medio de esta herramienta se debe tener en cuenta que se crean 2 clases predeterminadas: una desde $-\infty$ hasta el límite inferior y otra desde el límite superior de la última clase hasta $+\infty$. Por tanto, si se desea un histograma de m clases, se debe ingresar en el campo *Number* (interior) $m-2$ clases. En este caso en particular se desea construir un histograma de 15 clases; luego, el campo mencionado se debe llenar con el valor 13.

The screenshot shows the 'Histogram' dialog box with the following fields and annotations:

- Data File:** C:\Mis documentos\Si. *Annotation:* Directorio en el que se encuentra el archivo *.dat con la información. Éste se crea en la misma ubicación del modelo.
- Replications:** 1.
- Title:** Intervalo entre A y D. *Annotation:* Nombre de la gráfica.
- Axis Label:** Tiempo entre A y D. *Annotation:* Nombre del eje X.
- Histogram Cells:**
 - Number (Interior):** 13. *Annotation:* Número interior de celdas. Se tienen dos clases predeterminadas.
 - Lower Limit:** 13. *Annotation:* Límite inferior de todas las clases.
 - Width:** 4.16. *Annotation:* Ancho de clase del histograma.
- Save Cell Frequency in Files:**
 - Relative:** [Dropdown]. *Annotation:* Opciones para guardar las frecuencias relativas y acumuladas del histograma. Este archivo sólo se puede abrir a través del *Output Analyzer*.
 - Cumulative:** [Dropdown].

Gráfica del histograma

Intervalo entre A y D



Adicionalmente, se crea un resumen con la tabla de frecuencias, la cual se muestra a continuación:

Histogram Summary
Intervalo entre A y D

Cell	Cell Limits		Abs. Freq.		Rel. Freq.	
	From	To	Cell	Cumul	Cell	Cumul.
1	Infinity	12.9	0	0	0	0
2	12.9	17.06	18	18	0.07895	0.07895
3	17.06	21.22	22	40	0.09649	0.1754
4	21.22	25.38	36	76	0.1579	0.3333
5	25.38	29.54	23	99	0.1009	0.4342
6	29.54	33.7	18	117	0.07895	0.5132
7	33.7	37.86	18	135	0.07895	0.5921
8	37.86	42.02	12	147	0.05263	0.6447
9	42.02	46.18	19	166	0.08333	0.7281
10	46.18	50.34	7	173	0.0307	0.7588
11	50.34	54.5	8	181	0.03509	0.7939
12	54.5	58.66	6	187	0.02632	0.8202
13	58.66	62.82	10	197	0.04386	0.864
14	62.82	66.98	19	216	0.08333	0.9474
15	66.98	+Infinity	12	228	0.05263	1

En la anterior tabla se presentan los resultados de frecuencias observadas, relativas por clase y acumuladas. También se aprecian las dos clases predeterminadas, las cuales se estudiaron anteriormente, y las 13 que se incluyeron entre éstas para alcanzar un total de 15.

3.4 INTERVALO DE CONFIANZA PARA LA MEDIA



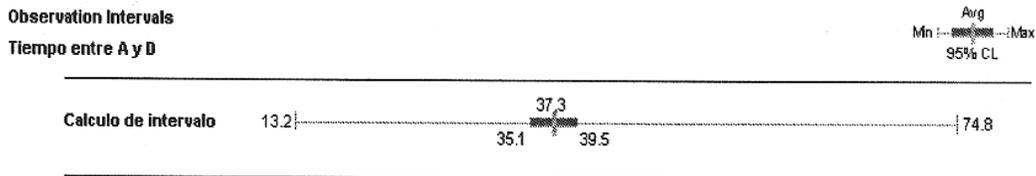
El analizador de datos también calcula, a un nivel de confianza específico, el intervalo de confianza para la media.

Botón para adición de datos (se pueden determinar varios intervalos de confianza a la vez). Aquí se busca el archivo *.dat que contiene los datos que se van a examinar. Estos archivos se generan en el mismo directorio del modelo

Título del gráfico

Nivel de confianza

En caso de que se quiera calcular varios intervalos de confianza a la vez, se muestran las gráficas con la misma escala (casilla activada). Esta característica resulta útil para visualizar *traslapos* entre intervalos de confianza



Adicionalmente, se genera un reporte como el que se muestra a continuación:

Classical C.I. Intervals Summary						
Tiempo entre A y D						
Identifier	Average Deviation	Standard Half-Width	0.950 C.I. Value	Minimum Value	Maximum Of Obs.	Number
Cálculo de intervalo	37.3	17	2.22	13.2	74.8	228

En el anterior reporte se muestran el valor mínimo y máximo, así como el promedio y la mitad del intervalo de confianza para la media de las observaciones recolectadas.

3.5 PROMEDIO ACUMULADO



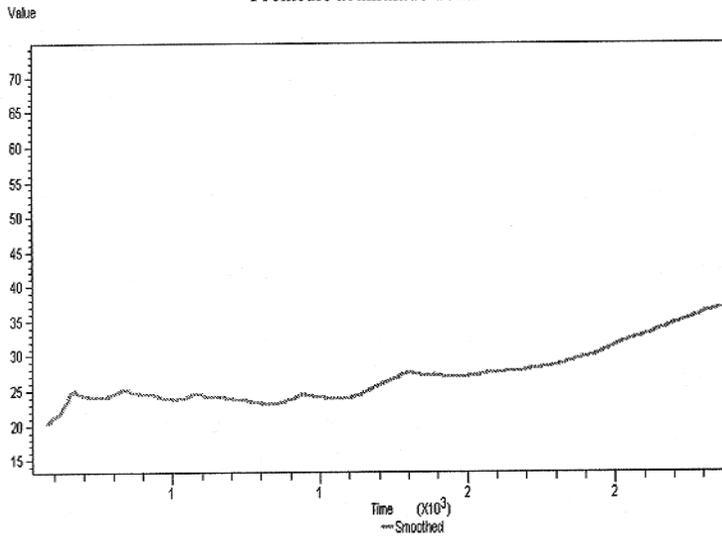
El promedio acumulado resulta de gran utilidad para determinar el período de estabilización del valor promedio de una variable de respuesta en el modelo; esto permite encontrar el tamaño de la corrida.

Directorio con el archivo *.dat que se va a analizar

Réplica a la que se le ha realizado el análisis. Si diferentes réplicas se pueden tomar en una gráfica (All) con los datos mezclados o se pueden graficar al mismo tiempo (Lumped)

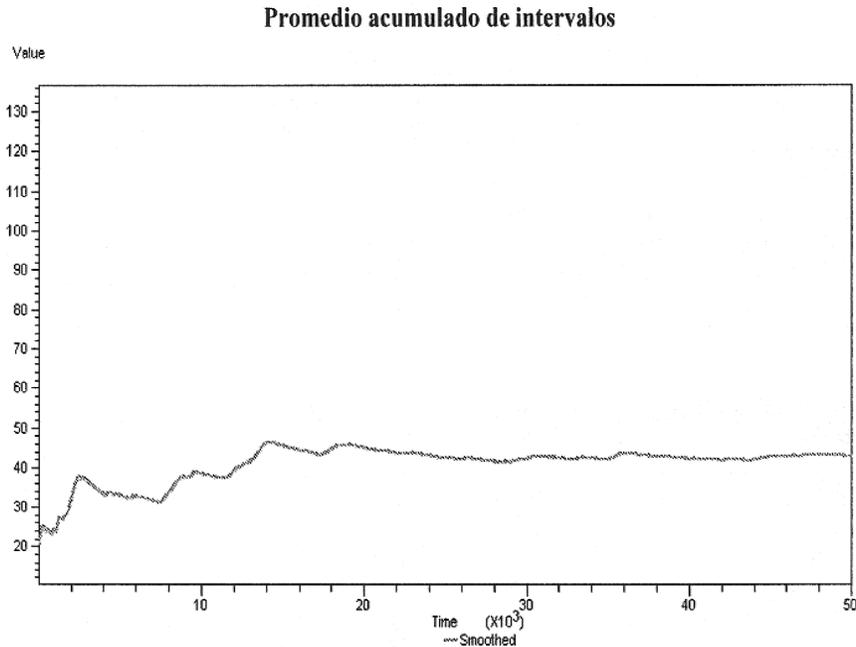
Tipo de promedio que se va a utilizar. En este caso es acumulado, pero puede ser móvil o suavización exponencial

Promedio acumulado de intervalos



La gráfica anterior muestra que, en el tiempo simulado (2.400 minutos), el promedio del tiempo que demora una entidad en ir de A hasta D no alcanza a ser un valor estable, por lo tanto, el tamaño de la corrida debe ser mayor.

Con el fin redeterminar el tamaño de la réplica, el modelo se corre en un tiempo de 50.000 minutos. La nueva gráfica de promedio acumulado se muestra a continuación:



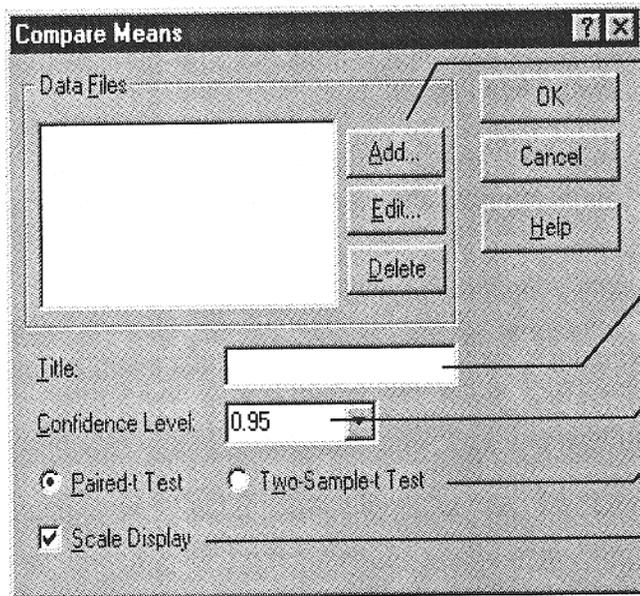
En la gráfica anterior se aprecia el período de transición de la variable hasta su estabilización en 20.000 minutos; por ello, éste será el valor mínimo para el tiempo de corrida.

3.6 COMPARACIÓN DE MEDIAS

Asuma que para el caso de la sección *Recolección de estadísticas de la entidad* (p. 159) se están probando dos métodos para el proceso C; el primero arroja la distribución que ya se conoce y el segundo ofrece una distribución de tiempo UNIF(4,14). Suponga que se desea determinar si no hay diferencia estadística entre los métodos en términos de la variable de respuesta, en este caso el tiempo entre A y D.

Este problema requiere de comparación de medias; para ello se corre el modelo y se almacenan los datos en un archivo *.dat, ejemplo, *Intervalo.dat*. Posteriormente se hacen las modificaciones necesarias y se corre nuevamente el modelo guardando los datos en un archivo diferente (cambiar el nombre del archivo), ejemplo, *Intervalo2.dat*.

Mediante el menú *Analyze/Compare Means* se ingresa a la opción de análisis de interés. El programa ofrece las opciones de comparación de medias muestrales o de muestras apareadas:



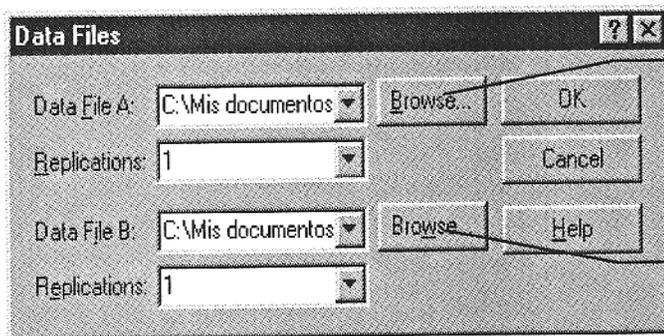
Botón de adición de pares de archivos con datos que se van a comparar

Título del gráfico

Nivel de confianza

Tipo de prueba

Ajustar pruebas a la misma escala. Esta opción es útil cuando se comparan varios pares de resultados al mismo tiempo



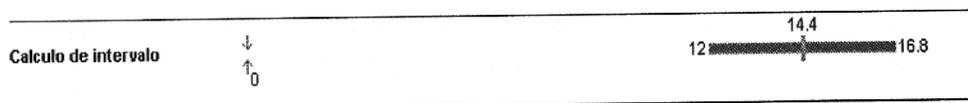
Directorio con el archivo *.dat de los datos correspondientes a la primera muestra de datos que se va a comparar

Directorio con el archivo *.dat de los datos correspondientes a la segunda muestra de datos que se va a comparar

A continuación se muestra la gráfica con el intervalo de confianza para el análisis:

Paired-t Comparison of Means
Comparación de medias

Diff
95% CL Test Value



En la anterior gráfica se aprecia que el cero no está contenido en el intervalo de confianza de las diferencias; por ello se puede concluir que las medias son diferentes.

A continuación se muestra la tabla resumen de la prueba *Paired-T Means Comparison* (Comparación de medias).

Identifier	Estd. Mean Difference	Standard Deviation	0.950 C.I. Half-Width	Minimum Value	Maximum Value	Number Of Obs
Cálculo de intervalo	14.4 9.03	18.4 42.3	2.41 227	13.2	74.8	227

REJECT H0 => MEANS ARE NOT EQUAL AT 0.05 LEVEL

3.7 ANÁLISIS DE VARIANZA (UN SÓLO FACTOR)

Suponga que se tiene un tercer método para el proceso C, el cual arroja una distribución exponencial con media de 8 minutos. Se desea probar si los diferentes métodos o tratamientos tienen efectos sobre la variable de respuesta, en este caso el tiempo entre A y D.

Se debe modificar el modelo con el fin de implementar el nuevo método, correrlo otra vez mediante el almacenamiento de los datos en un nuevo archivo *.dat, ejemplo, *Intervalo 3.dat*.

Se recurre al análisis de varianza mediante el menú *Analyze/One-way ANOVA*.

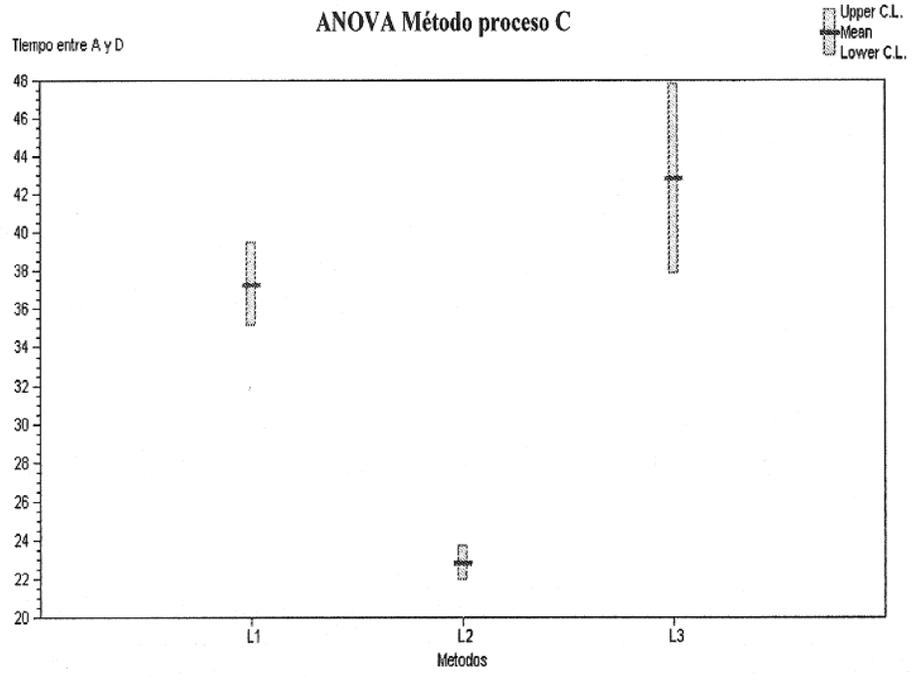
Importante. Si se van a desarrollar análisis de diferencias de medias entre tratamientos con la prueba de *Tukey*, el número de observaciones debe ser igual. Para esto los datos se deberán truncar o agrupar.

A continuación se muestra el cuadro de diálogo que permite la realización del análisis de varianza para experimentos de un solo factor:

The image shows a screenshot of the 'One-Way ANOVA' dialog box in a software application. The dialog box has a title bar with a question mark and a close button. It contains several sections: 'Data Files' with a list of files and buttons for 'Add...', 'Edit...', and 'Delete'; 'Comparison Method' with a dropdown menu set to '<No compare>'; 'Confidence Level' with a dropdown menu set to '0.95'; 'Title' with a text field containing 'VA Metodo proceso C'; 'X-Axis Label' with a text field containing 'Metodos'; and 'Y-Axis Label' with a text field containing 'Tiempo entre A y D'. On the right side, there are buttons for 'OK', 'Cancel', and 'Help'. Arrows point from text labels to specific elements in the dialog box.

- Botón de adición de archivos *.dat que tienen los resultados de los experimentos con diferentes niveles del factor controlable. En este caso se adicionan los archivos con los resultados para los tres métodos del proceso C (*Intervalo.dat*, *Intervalo2.dat* e *Intervalo3.dat*)
- Opción de comparación de medias. Se puede recurrir a los métodos de *Bonferroni*, *Scheffe* y *Tukey*
- Nivel de confianza de la prueba
- Título de la gráfica
- Nombre del factor controlable
- Nombre de la variable de respuesta

A continuación se presenta la gráfica de los intervalos de confianza para cada uno de los niveles:



INTERFAZ ARENA[®] – EXCEL[®]

En muchas ocasiones se requiere hacer un estudio o análisis más profundo de los resultados de la simulación de un modelo en ARENA[®]. Es cierto que este programa brinda, mediante reportes, las estadísticas de cada variable que entra en juego en el modelo simulado; sin embargo, no cuenta con una herramienta especializada con la que se pueda hacer, por ejemplo, una comparación entre los resultados de la simulación de un modelo respecto a otro, o una comparación entre los valores teóricos que se esperan del modelo (en caso de que éstos existan) y los resultados prácticos que se obtienen mediante la simulación.

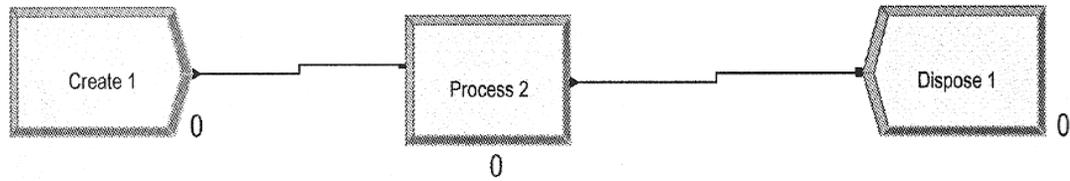
Para desarrollar estos estudios es necesario recurrir a otras herramientas comerciales, como MICROSOFT OFFICE[®]. En este capítulo se estudia la interacción entre ARENA[®] y las herramientas de este programa, y específicamente entre ARENA[®] y MICROSOFT[®] EXCEL[®].

A continuación se analizan, paso a paso, cómo se deben ejecutar los modelos que se han creado en ARENA[®] desde una hoja de cálculo de MICROSOFT[®] EXCEL[®], y cómo se deben capturar los resultados de la simulación para hacer el estudio deseado sobre esta hoja de cálculo.

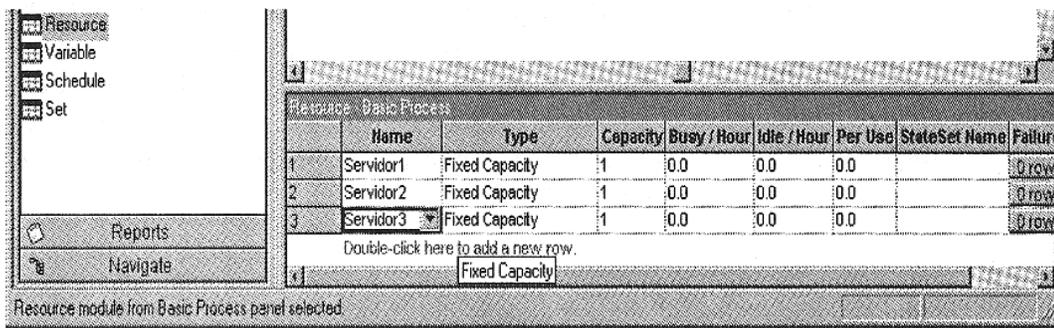
En un sistema MMK se identifican tres tipos de parámetros. El primero de ellos es la *tasa de llegada* (λ) de entidades al sistema; este parámetro representa el número de entidades por unidad de tiempo que llegan al sistema. El segundo es la *tasa de servicio* (μ), el cual, como su nombre lo indica, es el número de entidades por unidad de tiempo que un servidor atiende; es el mismo para todos los servidores. El tercer parámetro es el *número de servidores* (k) que posee el sistema para atender a las entidades. Para el caso que se estudia a continuación se han escogido tres servidores, $k=3$, es decir, se va a modelar un sistema MM3.

1. CREACIÓN DEL MODELO EN ARENA[®]

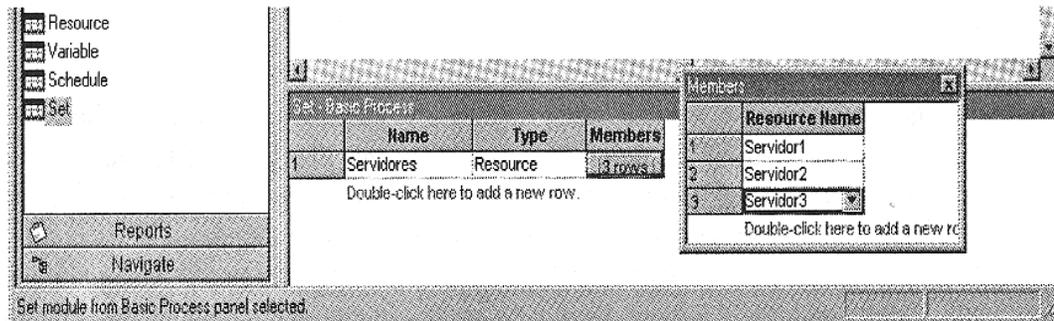
- Inicialmente se debe abrir un nuevo modelo en ARENA[®].
- Se insertan tres módulos del panel de procesos básicos: *Create*, *Process* y *Dispose*; luego se conectan. Este procedimiento se muestra a continuación:



- Se necesitan tres servidores en este modelo. Para desarrollar el modelo de estos servidores se crean tres recursos (cada servidor es un recurso utilizable del sistema). Luego se selecciona el módulo *Resource* y se adicionan tres recursos (*Servidor1*, *Servidor2*, *Servidor3*), así como se muestra en la siguiente figura:



- Mediante el módulo *Set* se adiciona un conjunto de recursos llamados *servidores*; luego los recursos *Servidor1*, *Servidor2* y *Servidor3* se adicionan como miembros (*Members*) de este conjunto.



- Ahora, dentro de la lógica del módulo *Process* se especifica que éste trabaje con el conjunto de servidores. Se hace doble clic en el bloque *Process* y se especifica como acción (*Action*) *Seize Delay Release*.

Process

Name: Process 1 Type: Standard

Logic

Action: Seize/Deplete/Release Priority: Medium(2)

Resources:

Resources: [Empty List] Add Edit Delete

Delay Type: Triangular Unit: Hours Allocation: Value Added

Minimum: 5 Value (Most Likely): 1 Maximum: 1.5

OK Cancel Help

- Mediante el botón *Add* se especifica que las entidades que lleguen a este bloque hagan uso de uno de los recursos (*Quantity=1*) que pertenecen al conjunto de servidores (*Type=Set, Set Name=Servidores*) si se encuentra disponible. Luego se escoge como regla de selección la *Preferred Order*, para que se escoja el primer recurso (servidor) que esté libre. Finalmente, se presiona *ok* y nuevamente *ok*.

Process

Name: Process 1 Type: Standard

Logic

Action: Resources

Seize Type: Set

Resource: Set

Set Name: Servidores Quantity: 1

Selection Rule: Preferred Order Save Attribute:

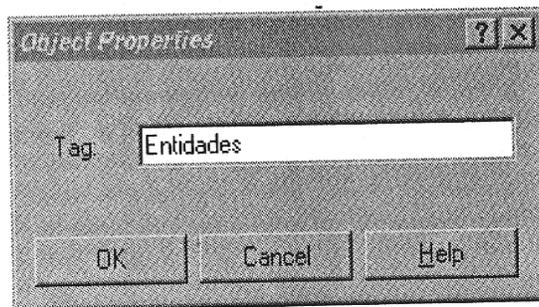
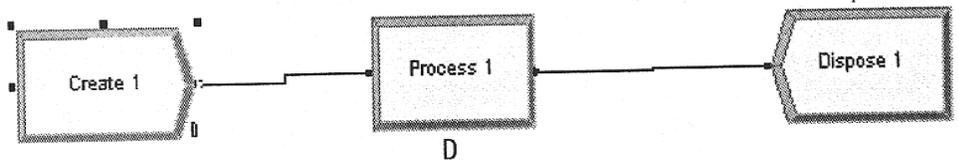
Delay: Triangular

Minimum: 5 Value (Most Likely): 1 Maximum: 1.5

OK Cancel Help

Los anteriores pasos se realizan en la parte de especificación de parámetros de cada uno de los módulos; los que se señalan a continuación se desarrollan desde EXCEL®.

- Para poder manipular cada uno de los módulos mediante código (ya sea desde ARENA® o EXCEL®), cada bloque tiene un identificador único (*Tag*). Un módulo se modifica al hacer clic derecho sobre él y al seleccionar la opción *Properties*. ARENA® muestra el *Tag* actual para el módulo seleccionado y permite modificarlo. Para el ejemplo en estudio, se colocan *Entidades* al *Tag* del módulo *Create* y *Proceso* al *Tag* del módulo *Process* (ver la siguiente figura).

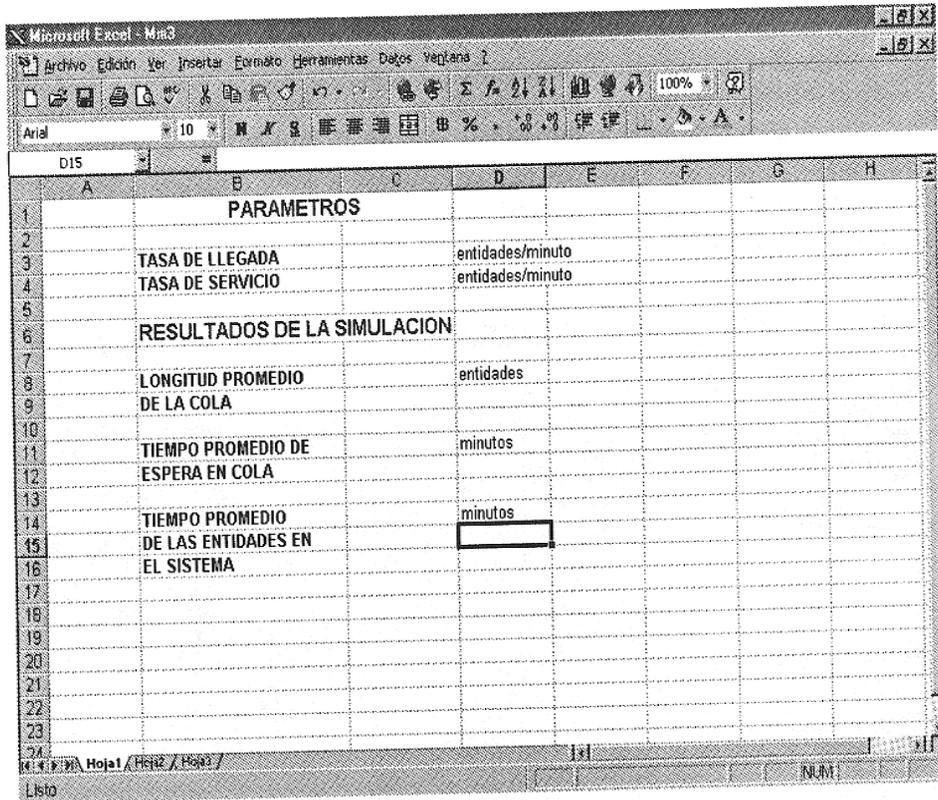


- Se guarda el modelo con el nombre *MM3.doc*.

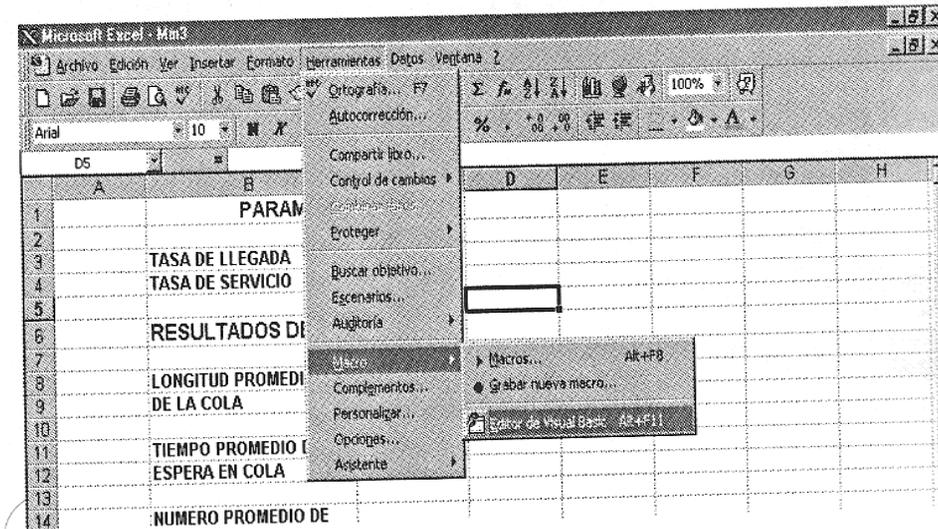
2. CREACIÓN DE LA HOJA DE CÁLCULO EN EXCEL®⁶

- Inicialmente se debe abrir un nuevo libro en Excel®.
- En una de las hojas se hace exactamente el formato que se muestra a continuación:

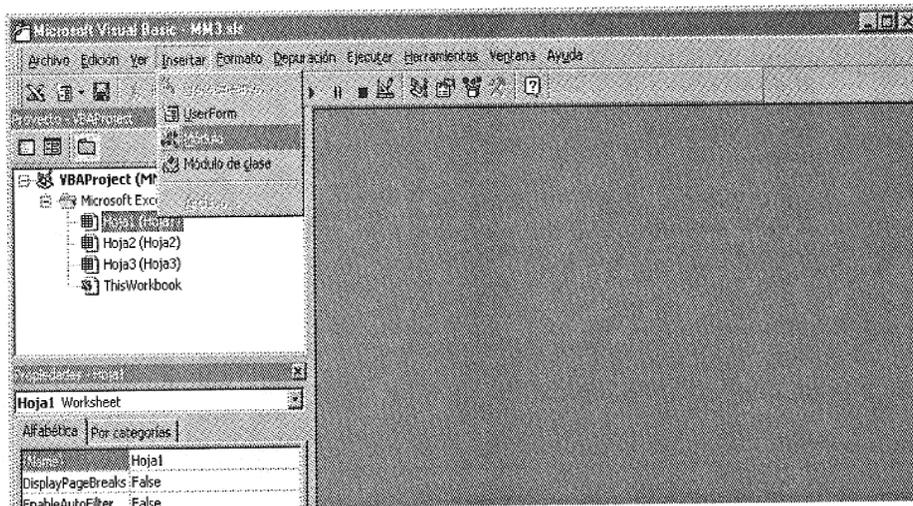
⁶ Todas las pantallas que se muestran en este apartado provienen de MICROSOFT EXCEL® 97. Cabe mencionar que no existe ningún problema con la implementación de estos ejemplos en versiones posteriores de EXCEL®.



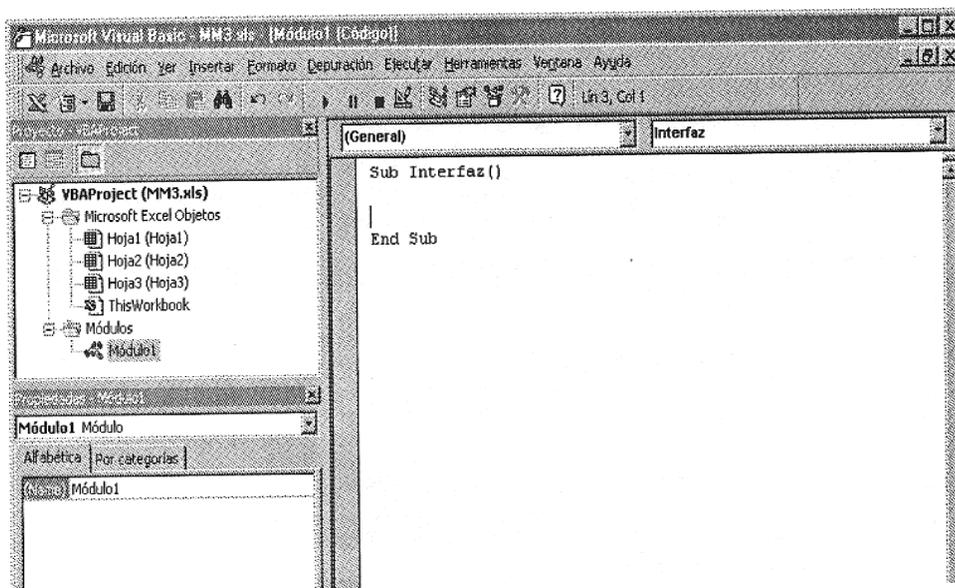
- Se guarda el libro con el nombre *MM3.xls*. Para el correcto funcionamiento del caso en estudio, los archivos de la hoja de cálculo y el modelo de ARENA[®] se deben encontrar en la misma carpeta.
- Se procede a ejecutar los primeros pasos de la *Interfaz*. Con este propósito se utiliza el *Visual Basic[®] para aplicaciones*, que se encuentra en todas las herramientas de MICROSOFT OFFICE[®].



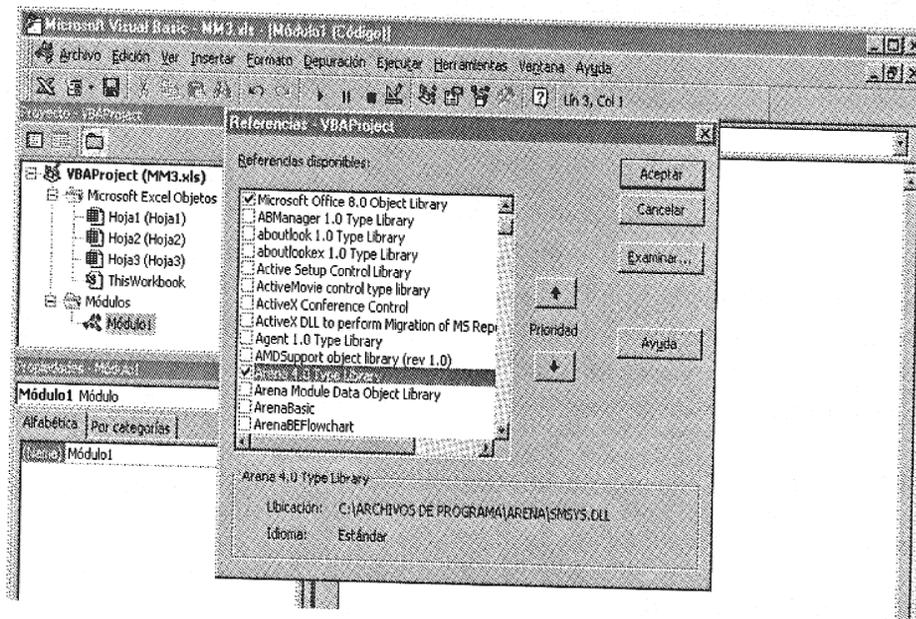
- Se abre el editor de *Visual Basic* (menú *Herramientas, Macro, Editor de Visual Basic* o la combinación de teclas *Alt-F11*).
- Cuando este editor se encuentre abierto, la opción *Módulo* en el menú *Insertar* se selecciona. La inserción de un módulo es necesaria para la creación de procedimientos o subrutinas definidas por el usuario.



- Dentro del módulo es preciso crear un procedimiento llamado *Interfaz*, que será el encargado de ejecutar el modelo que se ha creado en ARENA® para capturar los resultados de la simulación.



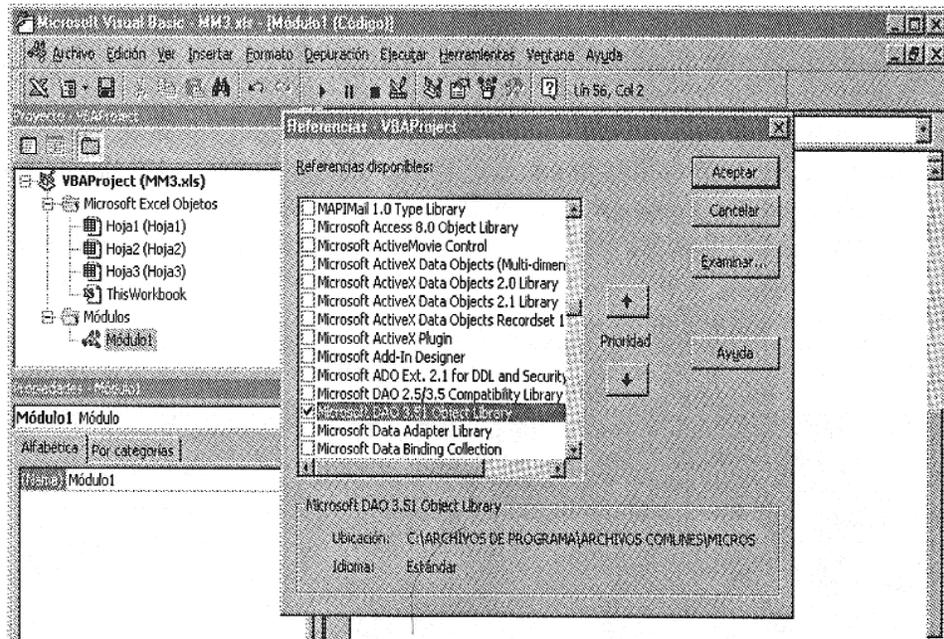
- Antes de comenzar a escribir el código, es preciso asegurarse de tener disponible la librería de objetos de ARENA[®], para poder manipular los modelos creados con esa herramienta. Por lo tanto, es necesario que se añada una referencia a la librería mencionada; para ello, en el menú de *Herramientas* se selecciona la opción *Referencias*. Aparecerá un listado con las librerías que se encuentran referenciadas actualmente y todas a las que se le pueda agregar una referencia. Se debe buscar la librería *Arena[®] Type Library* y seleccionarla. Posteriormente se hace clic en *Aceptar*. Una vez se realiza esta operación, se puede manipular la simulación de los modelos creados en ARENA[®].



La librería de ARENA[®] añadida brinda la posibilidad de ejecutar la aplicación *Arena.exe*, abrir un modelo creado para su simulación y modificar los parámetros de los módulos pertenecientes al modelo. Las operaciones anteriores se pueden hacer mediante las clases *Arena.Application*, *Arena.Model* y *Arena.Module*, respectivamente.

Cuando termina la simulación de un modelo, ARENA[®] genera un archivo de base de datos ACCESS[®] con la información (estadísticas) del modelo. Éste es el archivo que, una vez finalice la ejecución de la simulación, interesa consultar para extraer de él los resultados que sean de interés. Para eso se necesita también agregar una referencia a la librería de OBJETOS de ACCESO[®] a DATOS (DAO) de MICROSOFT[®].

- De la misma forma que se agrega la referencia a la librería de ARENA[®], se adiciona una referencia a *Microsoft[®] DAO 3.51 Object Library*. Esta librería brinda las clases para acceder a la base de datos (*DAO.Database*) y recuperar los valores de los registros almacenados en sus tablas (*DAO.Recordset*).



- Con estas dos referencias terminadas, ya se está preparado para escribir el código necesario para la creación de la *Interfaz*. A continuación se muestra todo el código del procedimiento *Interfaz* y se explica la función de cada una de sus partes.

Sub Interfaz()

Las siguientes tres líneas corresponden a la declaración de las variables objeto que pertenecen a la clase ARENA®.

Dim app As Arena.Application

Dim modelo As Arena.Model

Dim modulo As Arena.Module

La siguiente línea se refiere a la declaración de la variable objeto que sirve para manipular la hoja de cálculo de la cual provienen los parámetros de entrada y hacia la cual van los resultados.

Dim hoja As Excel.Worksheet

Las siguientes dos líneas corresponden a las declaraciones de las variables objeto que pertenecen a la clase DAO para el acceso a la base de datos en la cual ARENA® almacena los resultados de la simulación.

Dim bd As DAO.Database

Dim tabla As DAO.Recordset

Declaraciones de variables que se utilizarán posteriormente:

Dim ruta, criterio As String

Dim id As Long

Dim x, y As Single

Se hace que la variable hoja referencie a la hoja de cálculo en la cual se encuentra el formato de trabajo.

Set hoja = Hoja1

Se capturan los parámetros que se deben enviar al modelo en ARENA[®]. Los valores x y y corresponden a tasas de llegada y de salida de entidades, respectivamente. Para este tipo de parámetros la función de distribución exponencial es la apropiada, pero el tiempo entre llegadas o el tiempo entre salidas requieren de parámetro; por lo tanto se hace el cálculo de estos tiempos como el inverso de las respectivas tasas.

x = hoja.Cells(3, 3)

y = hoja.Cells(4, 3)

t1 = 1 / Val(x)

t2 = 1 / Val(y)

En una variable de cadena se guarda la ruta (path) en la cual se encuentra el libro.

ruta = ThisWorkbook.Path

Se ejecuta la aplicación Arena.exe de ARENA[®] y se abre el modelo creado anteriormente. Como se sabe que el archivo del modelo se encuentra en la misma carpeta que el libro de EXCEL[®], se utiliza el contenido de la variable ruta para ubicar el archivo.

Set app = CreateObject("Arena.Application")

Set modelo = app.Models.Open(ruta + "\MM3.doe")

ARENA[®] asigna a cada módulo del modelo un número único que identifica a cada bloque. Mediante el método Find de la clase Arena.Model.Modules se puede recuperar este número del módulo, especificando el Tag que lo identifica. Se busca el módulo Create, cuyo Tag es Entidades.

id = modelo.Modules.Find(smFindTag, "Entidades")

Se hace que la variable Módulo referencie al módulo Create del modelo, haciendo uso del número identificador del módulo.

Set modulo = modelo.Modules(id)

Mediante la propiedad Data(operando) de la clase Arena.Module se pueden modificar los parámetros de cada módulo. Sin embargo, hay que tener en cuenta que el parámetro Operando de la propiedad Data no corresponde, muchas veces, con el valor del parámetro que aparece en la caja de diálogo de cada módulo en ARENA[®]. Por ejemplo, para el parámetro Type del módulo Create, el nombre del operando respectivo es Interarrival Type. En la carpeta en la que se encuentra instalado ARENA[®] en el computador se ubican unos archivos de texto (BasicProcess.txt, Advanced Process.txt, AdvancedTransfer.txt) en los que se especifican los nombres de los operandos para la propiedad Data para todos los parámetros de cada uno de los módulos pertenecientes a cada uno de los paneles de ARENA[®].

Se modifica el parámetro Type del módulo Create.

modulo.Data("Interarrival Type") = "Expression"

Se modifica el parámetro Expression del módulo Create. Con la concatenación se quiere ingresar una expresión válida para ARENA[®] como EXPO(0.33). Es preciso tener en cuenta que la distribución exponencial es la apropiada.

modulo.Data("Expression") = "EXPO(" & Str(t1) & ")"

Se especifica el parámetro Units del módulo Create para determinar las unidades de tiempo en que se consideran los tiempos entre llegadas sucesivas.

modulo.Data("Units") = "Minutes"

Se realiza el mismo procedimiento con el módulo Process, al que se le había asignado el Tag Proceso.

id = modelo.Modules.Find(smFindTag, "Proceso")
Set modulo = modelo.Modules(id)

modulo.Data("DelayType") = "Expression"
modulo.Data("Expression") = "EXPO(" & Str(t2) & ")"
modulo.Data("Units") = "Minutes"

Es preciso asegurarse de generar el archivo (base de datos) con las estadísticas del modelo.

modelo.DisableReportDatabase = False

Para evitar que salga al final de cada réplica un mensaje que pregunte si se desea ver el reporte de ARENA[®], se coloca la siguiente línea:

modelo.DisplayDefaultReport = smNeverDisplay

Con las siguientes líneas se especifica la unidad de tiempo base para el modelo (aquella con la que se regirá el reloj de la simulación y en la que se darán los resultados en la base de datos), el número de réplicas que se desean, la longitud de cada réplica, las unidades de tiempo en que se mide esta longitud y la velocidad de la simulación.

modelo.BaseTimeUnits = smMinutes
modelo.NumberOfReplications = 1
modelo.ReplicationLength = 1
modelo.ReplicationLengthTimeUnits = smHours
modelo.RunSpeed = 10

Se inicia la ejecución del modelo y se hace que Visual Basic no ejecute ninguna línea de código hasta que el modelo no termine su ejecución. Una vez termine la réplica, se le envía una orden End al modelo para que también se le ponga fin a la simulación.

modelo.Go smGoWait

modelo.End

Se guardan los cambios realizados y se cierra el modelo.

modelo.Save

modelo.Close

Se realiza la conexión con la base de datos. ARENA[®] siempre genera la base de datos de reporte en el mismo directorio en que se encuentre el archivo del modelo. En este caso en particular es el mismo directorio en el que se localiza el libro de EXCEL[®] y el mismo que se encuentra almacenado en la variable ruta.

Set bd = Workspaces(0).OpenDatabase(ruta + "\mm3.mdb")

Dentro de los objetos de la base de datos hay una consulta o vista llamada StatsAndOutputQry, la cual contiene todas las estadísticas coleccionadas por ARENA[®] durante la simulación. Los campos primordiales en esta vista son: Value (valor), AvgObs (promedio observado), MaxObs (valor máximo observado), MinObs (valor mínimo observado), SourceProcess.Name (objeto-entidad, recurso, cola, etc. sobre el que se generó la estadística) y SourceDataType.Name (propiedad observada). Para una mejor comprensión de lo expuesto es recomendable que se abra la base de datos con MICROSOFT ACCESS[®], con el propósito de que se examinen estos campos y se decida qué pareja de valores de los

campos SourceProcess.Name y SourceDataType.Name es la que corresponde a la estadística que se desea recuperar.

En la siguiente línea se muestra la cadena SQL con la que se capturan los datos almacenados en la base de datos de la vista mencionada.

```
Sql = "Select Value as valor, AvgObs as promedio, MaxObs as maximo, "+ _  
      " MinObs as minimo, SourceProcess.Name as objeto, "+ _  
      " SourceDataType.Name as propiedad " + _  
      " From StatsAndOutputQry"
```

Se hace la consulta a la base de datos; el resultado de ésta queda almacenado en la variable tabla.

```
Set tabla = bd.OpenRecordset(Sql, dbOpenDynaset)
```

Se verifica que sí se hayan recuperado registros.

```
If Not tabla.EOF Then
```

Se hace un movimiento hacia el inicio del conjunto de registros recuperados.

```
tabla.MoveFirst
```

Mediante una cadena de condición SQL se especifican los valores de la pareja objeto-propiedad (SourceProcess.Name-SourceDataType.Name) de la estadística que se desea recuperar. En este caso se trata del número promedio de personas en la cola. Por eso, el objeto sobre el que se generó la estadística fue Process 1.Queue, la propiedad, Number Waiting, y el dato que se desea es el valor promedio.

```
criterio = "objeto='Process 1.Queue' and propiedad='Number Waiting'"
```

```
tabla.FindFirst criterio
```

Se guarda la estadística en la hoja de cálculo, específicamente en la celda que ha sido destinada para este fin.

```
hoja.Cells(8, 3) = Str(tabla.Fields("promedio"))
```

Como se mencionó anteriormente, el proceso de selección de los valores adecuados para los campos objeto y propiedad sólo es satisfactorio si se consulta directamente la base de datos y se analizan los valores que ARENA® ha almacenado en ella.

El procedimiento para recuperar los valores de cada estadística es el mismo.

```
tabla.MoveFirst
```

Se desea recuperar el tiempo de espera en cola.

```
criterio = "objeto='Process 1.Queue' and propiedad='Waiting Time'"
```

```
tabla.FindFirst criterio
```

```
hoja.Cells(11, 3) = Str(tabla.Fields("promedio"))
```

```
tabla.MoveFirst
```

Se desea recuperar el tiempo total de las entidades en el sistema.

```
criterio = "objeto='ENTITY 1' and propiedad='Total Time'"
```

```
tabla.FindFirst criterio
```

```
hoja.Cells(14, 3) = Str(tabla.Fields("promedio"))
```

```
End If
```

Se cierra el conjunto de registros y la base de datos.

tabla.Close

bd.Close

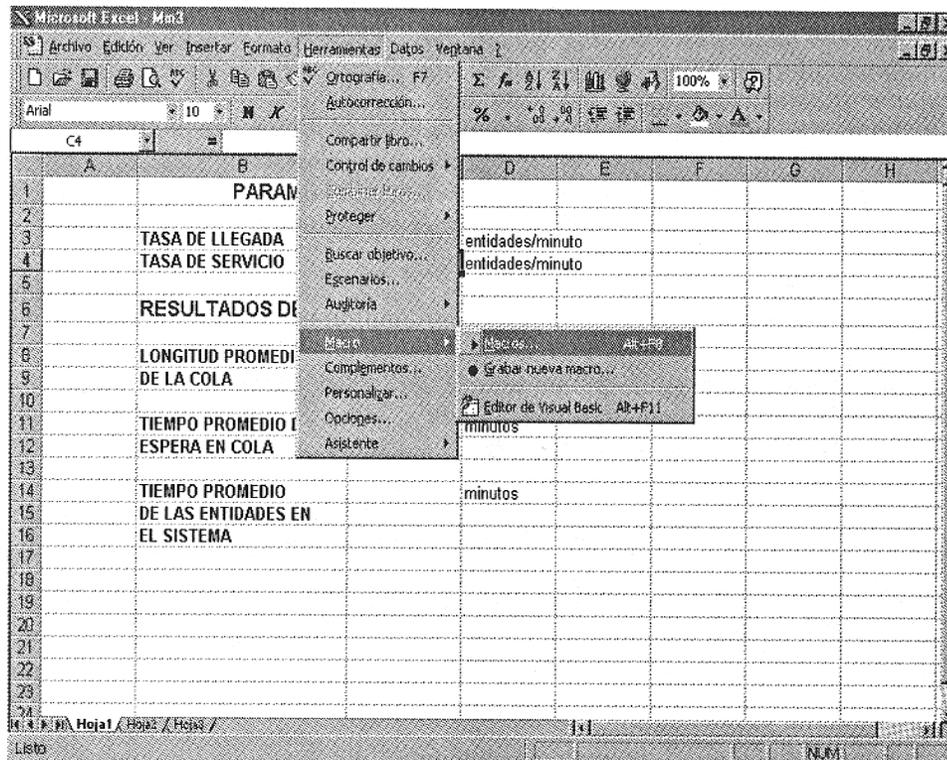
End Sub

El procedimiento no cambia en caso de que el modelo contenga módulos que utilicen contadores (*tallies*), o que el cálculo de algunas estadísticas adicionales se hayan especificado (mediante el módulo *Statistic*, por ejemplo). Estas estadísticas también quedan almacenadas en la base de datos. El único paso previo a la captura de estos resultados desde una hoja de cálculo en EXCEL® es abrir la base de datos e identificar bajo qué parejas (*SourceProcess.Name* y *SourceDataType.Name*) se han almacenado los resultados observados durante la simulación. Este proceso de identificación es bastante sencillo, ya que ARENA® utiliza los mismos nombres de módulos, contadores, *tallies* y estadísticas que el usuario especifica en el modelo.

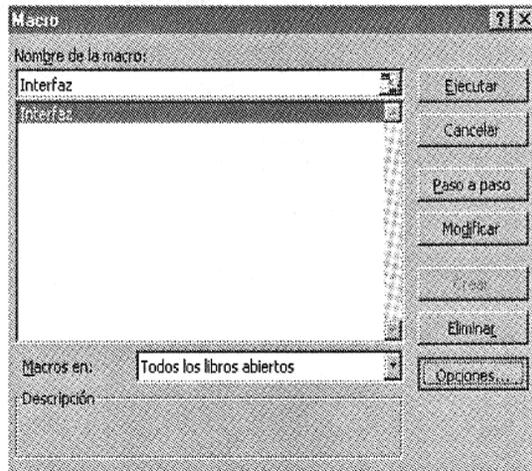
Una vez termine de escribir todo el código, se guardan los cambios realizados, se cierra la ventana del editor de *Visual Basic* y se regresa a la ventana tradicional de EXCEL®.

3. EJECUCIÓN DE LA INTERFAZ

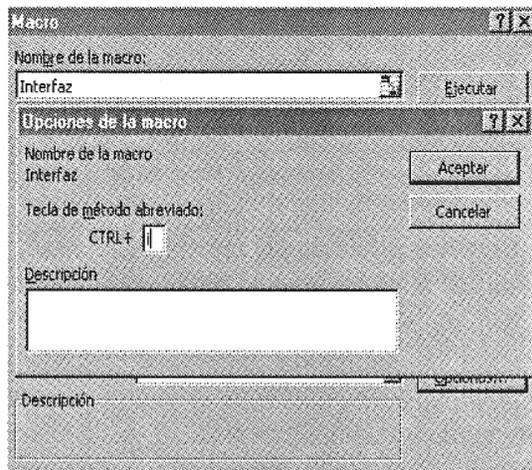
- Primero se asigna una secuencia de teclas a la ejecución de la simulación desde EXCEL®. Para realizar esta operación se selecciona la opción *Macros...* del menú *Herramientas, Macro*.



- Aparece una ventana con el listado de los procedimientos que se han creado (en este caso uno solo). Se hace un clic sobre el procedimiento *Interfaz* y luego se presiona el botón *Opciones...*



- La combinación de teclas *Ctrl+I* (o la que se desee) se asigna como tecla de método abreviado al procedimiento.



- Ahora que se tiene todo listo, se puede poner en práctica el funcionamiento de la *Interfaz*. Lo primero es suministrar, a la hoja de cálculo, las tasas de llegada de entidades al sistema y la tasa de servicio de cada uno de los servidores.

The screenshot shows a Microsoft Excel spreadsheet with the following content:

	A	B	C	D	E	F	G	H
1	PARAMETROS							
2								
3		TASA DE LLEGADA	1	entidades/minuto				
4		TASA DE SERVICIO	2	entidades/minuto				
5								
6	RESULTADOS DE LA SIMULACION							
7								
8		LONGITUD PROMEDIO		entidades				
9		DE LA COLA						
10								
11		TIEMPO PROMEDIO DE		minutos				
12		ESPERA EN COLA						
13								
14		TIEMPO PROMEDIO		minutos				
15		DE LAS ENTIDADES EN						
16		EL SISTEMA						
17								
18								
19								
20								
21								
22								
23								
24								

- Luego se presiona la secuencia de teclas que se acaba de asignar al procedimiento (*Ctrl+I*). EXCEL[®] comenzará a ejecutar el procedimiento; una vez termine, éste llenará la hoja de cálculo con los resultados observados durante la simulación del modelo. El proceso debe ser completamente independiente, es decir, el usuario no debe intervenir en ningún momento entre el comienzo de la ejecución de la *Interfaz* y la captura de los resultados.
- Una vez se termina la ejecución del modelo en ARENA[®], el estado de la hoja de cálculo cambiará de la manera como se muestra en la siguiente figura:

Anexo

DISTRIBUCIONES ESTADÍSTICAS

ARENA[®] posee una amplia gama de funciones o distribuciones estadísticas incorporadas para la generación de números aleatorios. Estas distribuciones aparecen cuando, en varios módulos en los que éstas se puedan necesitar, se hace clic en alguna de las listas desplegables de los menús. En este apartado se describen todas las distribuciones de ARENA[®].

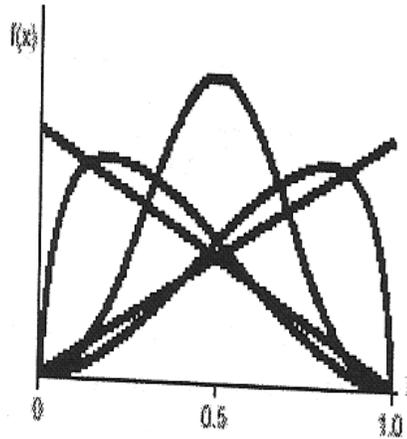
Cada distribución en ARENA[®] tiene sus propios parámetros asociados. Para poder especificar una distribución se deben introducir los valores a todos los parámetros. El número, el significado y el orden de los valores de los parámetros dependen de la distribución utilizada. A continuación se presenta un pequeño resumen de las distribuciones que posee ARENA[®] y de sus respectivos parámetros.

Distribución		Parámetros
Beta	BETA	Beta, Alfa
Continuous	CONT	CumP ₁ , Val ₁ , ..., CumP _n , Val _n
Discrete	DISC	CumP ₁ , Val ₁ , ..., CumP _n , Val _n
Erlang	ERLA	Media, k
Exponential	EXPO	Media
Gamma	GAMM	Beta, Alfa
Johnson	JOHN	Gamma, Delta, Lambda, Xi
LogNormal	LOGN	Media, LogStd
Normal	NORM	Media, Desviación Estándar
Poisson	POIS	Media
Triangular	TRIA	Mínimo, Media, Máximo
Uniform	UNIF	Mínimo, Máximo
Weibull	WEIB	Beta, Alfa

Para ingresar una distribución en ARENA[®] (cuando no aparezca una lista desplegable) se debe introducir la abreviatura correspondiente a cada distribución seguida de los parámetros encerrados entre paréntesis. Se pueden dejar espacios entre los parámetros para hacer que la lectura de éstos sea más fácil.

Beta

Función de densidad de probabilidad



Parámetros

ALFA y BETA; estos deben ser números reales positivos.

Rango

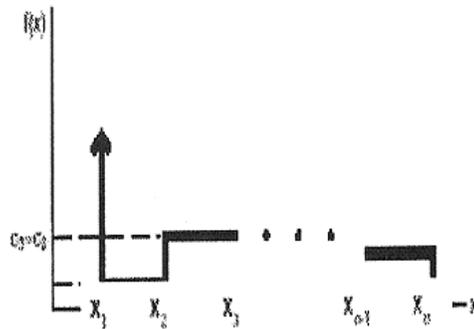
$[0,1]$. Éste se puede transformar a un rango $[a,b]$; posteriormente se explica esta operación.

Aplicaciones

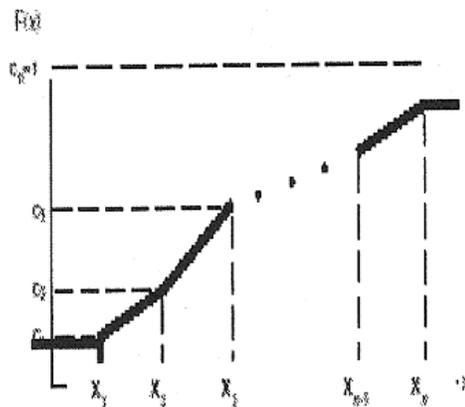
Debido a su habilidad para adaptarse a un gran número de formas, esta distribución se usa frecuentemente cuando hay ausencia de datos, ya que el rango de la distribución BETA va desde 0 hasta 1. La muestra X se puede transformar a la escala Beta Y a través de la siguiente ecuación: $Y=a+(b-a)X$. Esta distribución se usa frecuentemente para representar proporciones aleatorias, como la proporción de elementos defectuosos dentro de una producción.

Continuous (Continua)

Función de
Densidad de
probabilidad



Función de
distribución
acumulada



Parámetros

La función *Continuos* en ARENA[®] devuelve una muestra de una distribución que define el usuario. Las parejas formadas por las probabilidades acumuladas (*CumP*) y los valores asociados (*Val*) se necesitan especificar. La muestra que retorna será un número real comprendido entre x_1 y x_n .

Rango

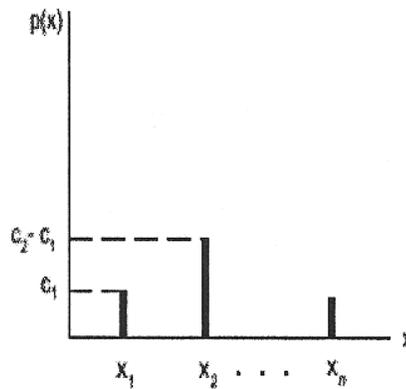
$[x_1, x_n]$

Aplicaciones

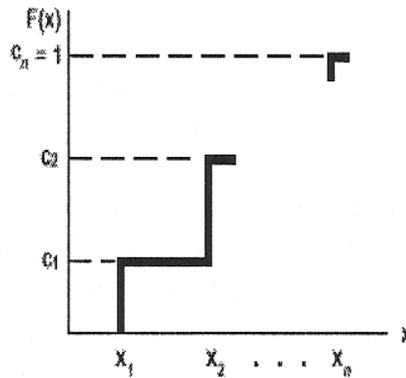
La distribución empírica se utiliza frecuentemente para agregar de manera directa al modelo de datos actuales variables aleatorias continuas. Esta distribución se puede usar como una alternativa para una distribución teórica cuando se tienen los valores que se acomodan a los datos.

Discrete (Discreta)

Función de Probabilidad



Función de distribución acumulada



Parámetros

La función *Discrete* en ARENA[®] devuelve una muestra de una distribución de probabilidad discreta definida por el usuario. La distribución se define por el conjunto de los n posibles valores discretos (denotados por x_1, x_2, \dots, x_n), que pueden ser retornados por la función y las probabilidades acumuladas (denotadas por c_1, c_2, \dots, c_n) que están asociadas con estos valores. La probabilidad acumulada c_j para x_j se define como la probabilidad de obtener un valor que es menor o igual a x_j y por esta razón se puede decir que las sumas de las $p(x_k)$ para k variando entre 1 y j . Por definición, c_n es igual a 1.

Rango

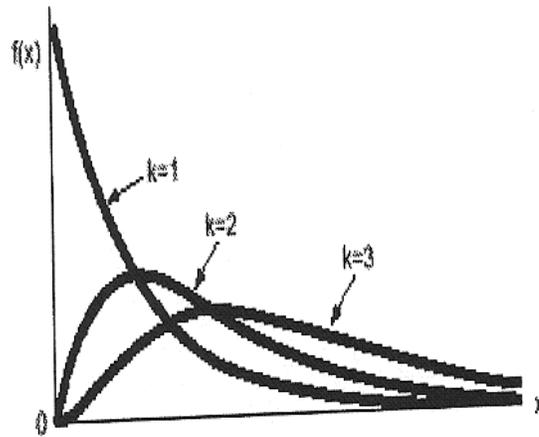
$\{x_1, x_2, \dots, x_n\}$

Aplicaciones

Esta distribución se usa frecuentemente para asignar una variable o atributo a un conjunto de valores que se basan en probabilidad. Por ejemplo, la fórmula $\text{DISCRETE}(0.25, 1, 0.6, 2, 1, 3)$ significa que el 25% pertenecerá a 1, el 35% ($35\% + 25\% = 60\%$) pertenecerá a 2 y el 40% restante, a 3.

Erlang

Función de densidad de probabilidad



Parámetros

Si x_1, x_2, \dots, x_n son variables aleatorias exponencialmente distribuidas de manera independiente e idéntica, entonces, la suma de estas n muestras tiene una distribución *Erlang* k . La media (β) de cada uno de los componentes y el número exponencial de variables aleatorias (k) son los parámetros para esta distribución. La media exponencial se debe definir como un número entero positivo real, y k se define como un entero positivo.

Rango

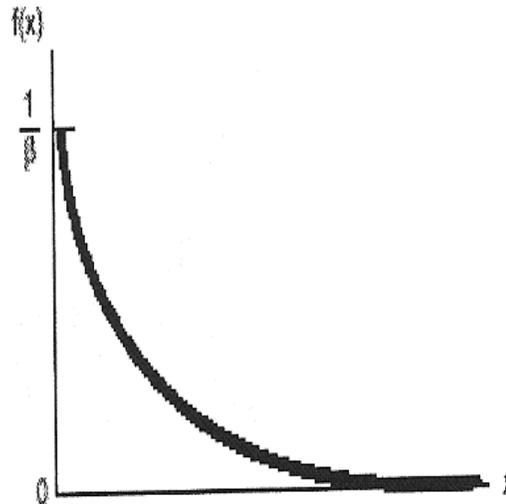
$[0, +\infty)$

Aplicaciones

Erlang se usa en situaciones en las que las actividades toman lugar en fases sucesivas y cada fase tiene una distribución exponencial. Para un número grande de k , *Erlang* se aproxima a la distribución normal. Se usa frecuentemente para representar el tiempo requerido para completar una labor; esta distribución es un caso especial de la distribución *Gamma*, en la cual el parámetro α de la *Gamma* es el entero k de la *Erlang*.

Exponencial

**Función de
distribución de
densidad**



Parámetros

La media (β) se debe definir como un número real positivo.

Rango

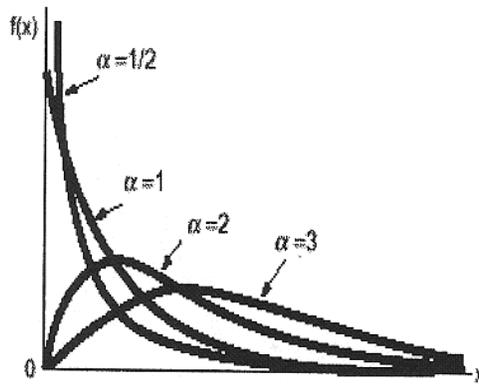
$[0, \infty)$

Aplicaciones

Esta distribución se usa frecuentemente para modelar tiempos entre eventos en llegadas al azar y procesos de descomposición, pero generalmente es inapropiada para modelar tiempo de retardos de procesos. En el módulo *Create* de ARENA[®] la opción horario (*Schedule*) extrae automáticamente una muestra de una distribución exponencial con una media que cambia de acuerdo con el horario definido. Esto es particularmente útil en aplicaciones de servicios en las cuales el volumen de los clientes varía a lo largo del día.

Gamma

Función de densidad de probabilidad



Parámetros

Los valores tanto de α como de β deben ser definidos por valores reales positivos.

Rango

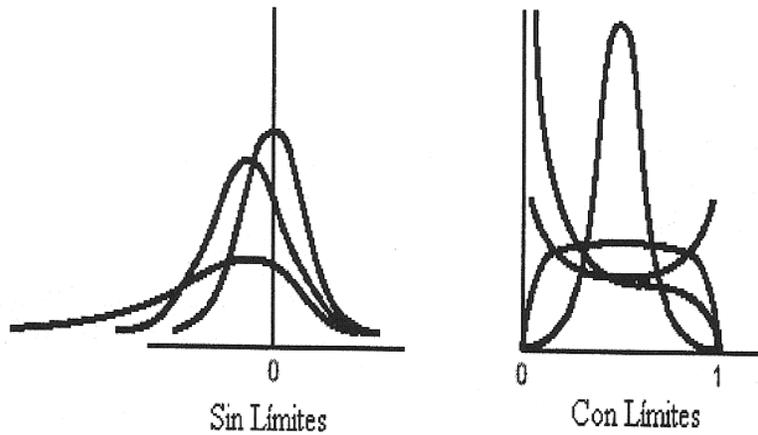
$[0, +\infty)$

Aplicaciones

Para valores enteros de α , la distribución *Gamma* es la misma *Erlang*. Esta distribución se usa frecuentemente con el fin de representar el tiempo que se requiere para completar cualquier trabajo.

Johnson

Función de densidad de probabilidad



Parámetros

El parámetro de forma (γ), los parámetros *Delta* ($\delta > 0$) y *Lambda* ($\lambda > 0$) y el parámetro de ubicación ξ (ξ).

Rango

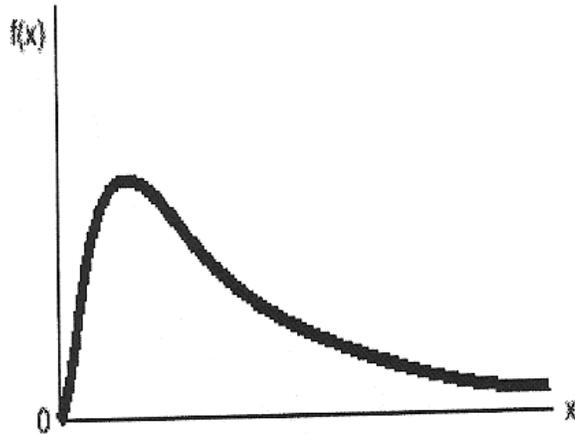
$(-\infty, +\infty)$ cuando no tiene límites.
 $[\xi, \xi + \lambda]$ cuando tiene límites.

Aplicaciones

La flexibilidad de la distribución *Johnson* permite ajustar muchos conjuntos de datos. ARENA[®] puede simular ambas clases de estilo (con y sin límites), y para decidir cuándo se quiere usar una y cuándo se desea usar otra, simplemente se debe alterar el parámetro *Delta* (δ). En caso de que el valor de este parámetro sea positivo, se usará la forma con límites; en caso contrario ($\delta < 0$), se escogerá la forma sin límites.

LogNormal

Función de densidad de probabilidad



Parámetros

El parámetro de escala (μ) se debe especificar como un número real y el parámetro de forma (σ), como un número real positivo.

Rango

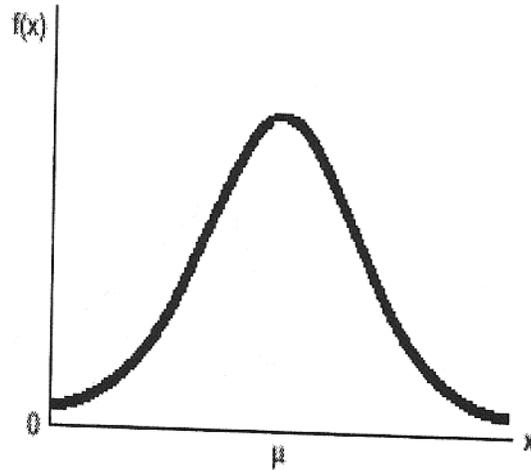
$[0, +\infty)$

Aplicaciones

LogNormal se usa en situaciones en las cuales la cantidad es el producto de un largo número de cantidades aleatorias. También se utiliza frecuentemente para representar los tiempos de trabajo de una distribución sesgada a la derecha. *LogNormal* se relaciona con la distribución *Normal*.

Normal

Función de densidad de probabilidad



Parámetros

La media (μ) y la desviación estándar (σ) deben ser números reales positivos.

Rango

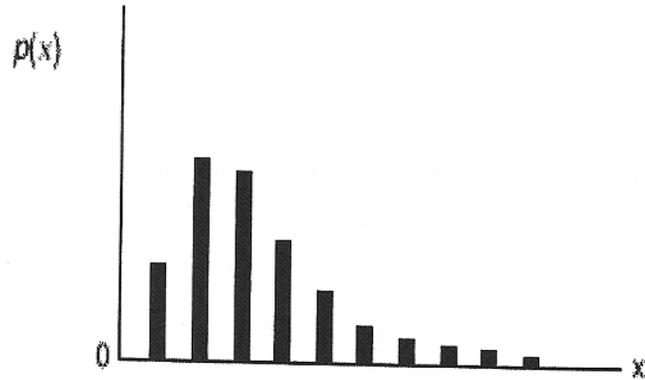
$(-\infty, +\infty)$

Aplicaciones

Normal se utiliza en situaciones en las cuales se aplica el teorema del límite central. También se usa empíricamente en muchos procesos que aparentan tener una distribución simétrica. Como su rango teórico fluctúa entre $-\infty$ y $+\infty$, esta distribución solamente se puede usar para cantidades positivas.

Poisson

Distribución de probabilidad



Parámetros La media (λ) se define como un número real positivo.

Rango $\{0, 1, \dots\}$

Aplicaciones *Poisson* es una distribución discreta que se usa frecuentemente para modelar un número aleatorio de eventos que ocurren en un intervalo de tiempo determinado. Si el tiempo entre eventos sucesivos se distribuye exponencialmente, entonces, el número de eventos que ocurren en un determinado intervalo de tiempo tiene una distribución de *Poisson*. Ésta se utiliza también para modelar agrupaciones de tamaños variables.