

MODELADO Y SIMULACIÓN

Febrero de 2014 - Segunda semana

PREGUNTA 1 (3 puntos)

Se pretende estudiar mediante simulación el sistema de admisión de clientes de una compañía aérea en un aeropuerto. La finalidad del sistema es proporcionar a los clientes su tarjeta de embarque y realizar la facturación del equipaje de los clientes. El funcionamiento del sistema se describe a continuación.

El sistema comienza a funcionar dos horas y media antes de la hora de salida del vuelo, y termina de funcionar cuarenta y cinco minutos antes de la hora de salida del vuelo. Durante el tiempo de funcionamiento del sistema, se produce la llegada de los clientes. Los clientes llegan en grupos, cuyo número de integrantes obedece a la distribución de probabilidad siguiente:

Personas por grupo	Probabilidad
1	0.15
2	0.45
3	0.15
4	0.15
5	0.10

El tiempo que transcurre entre la llegada de dos grupos consecutivos está distribuido exponencialmente, con media un minuto.

Algunos clientes ya disponen de tarjeta de embarque en el momento en que llegan al sistema, mientras que otros no la tienen. Se supone que todos los integrantes de un grupo se encuentran en la misma situación. Es decir, o bien todos disponen de tarjeta de embarque, o bien ninguno de ellos la tiene. La probabilidad de que un grupo tenga tarjeta de embarque es 0.25, mientras que la probabilidad de que no la tenga es 0.75.

Si el grupo ya dispone de las tarjetas de embarque, se dirige directamente a los mostradores de facturación. Si no dispone de tarjetas de embarque, primero deben adquirir las tarjetas y luego ir a los mostradores de facturación.

Los grupos que no disponen de tarjeta de embarque deben obtenerla. Las tarjetas de embarque son emitidas por máquina expendedoras. Hay 4 máquinas expendedoras, que funcionan independientemente entre sí. Si cuando el grupo de personas llega las máquinas están ocupadas, entonces esperan en una única cola con disciplina FIFO. Todas las personas del grupo son atendidas por una misma máquina expendedora, quien proporciona a todas ellas las tarjetas de embarque. El tiempo necesario para que todos los componentes de un grupo introduzcan sus datos en la máquina y ésta imprima las tarjetas de embarque es igual al número de integrantes del grupo multiplicado por un número aleatorio distribuido triangularmente, con rango $[1, 3]$ minutos y moda 2 minutos.

Tanto los grupos que al llegar al sistema ya disponían de tarjeta de embarque, como aquellos que acaban de obtenerla en las máquinas expendedoras, deben esperar una cola con disciplina FIFO delante de los mostradores de facturación. Hay 2 mostradores de facturación, que funcionan independientemente entre sí. Se forma una única cola frente a los mostradores.

Cuando llega su turno, el grupo completo se dirige a uno de los mostradores. El tiempo necesario para facturar el equipaje del grupo es igual al número de personas del grupo multiplicado por un número aleatorio distribuido uniformemente entre 0.5 y 1 minutos. Una vez han sido atendidos en el mostrador de facturación, el grupo de clientes abandona el sistema.

Obsérvese que cada grupo de personas se mueve por el sistema sin dividirse: el grupo completo dispone o no de tarjeta de embarque, el grupo completo es atendido por una máquina (si procede), el grupo completo es atendido en un mostrador y el grupo completo abandona el sistema.

El objetivo del estudio es estimar el número de grupos que quedan sin atender en el instante en que el sistema termina de funcionar. Para ello, se realizan 1000 réplicas independientes de la simulación de los 105 minutos de funcionamiento del sistema.

Describe *detalladamente* cómo realizaría el modelo del sistema anterior usando Arena. En particular, dibuje el diagrama de módulos e indique qué parámetros del comportamiento del sistema deben definirse en cada módulo.

Solución a la Pregunta 1

En la Figura 1.1 se muestra una posible manera de describir el funcionamiento del sistema.

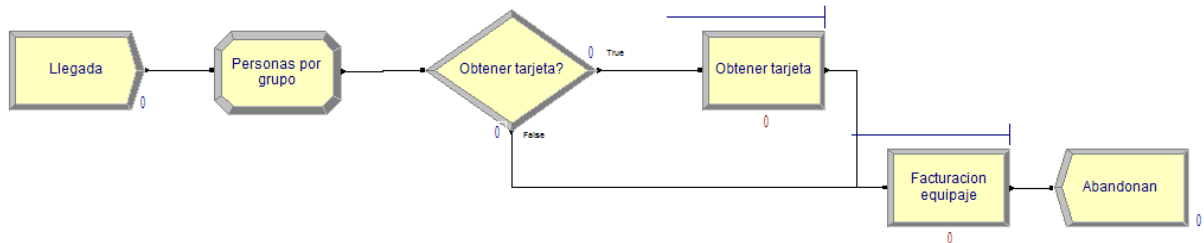


Figura 1.1: Diagrama de módulos.

El proceso de llegada de los viajeros puede ser descrito tal como se muestra en la Figura 1.2. Obsérvese que la entidad es el grupo. Se asocia a cada grupo un atributo, que es el número de personas de que consta el grupo. Dicha asociación se realiza en el módulo *Assign* (véase la Figura 1.3)

Se emplea un módulo *Decide* para bifurcar el flujo de las entidades, dependiendo de si el grupo de pasajeros tiene o no tarjeta de embarque. El módulo *Decide* se muestra en la Figura 1.4.

La definición de los dos procesos es mostrada en las Figuras 1.5 y 1.6. La capacidad de los recursos se fija en el módulo de datos *Resource* (véase la Figura 1.7).

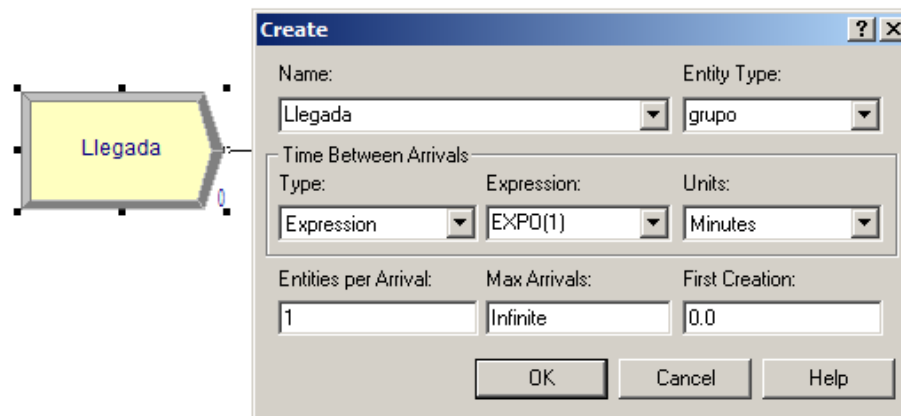


Figura 1.2: Módulo *Create* que describe la llegada de grupos.

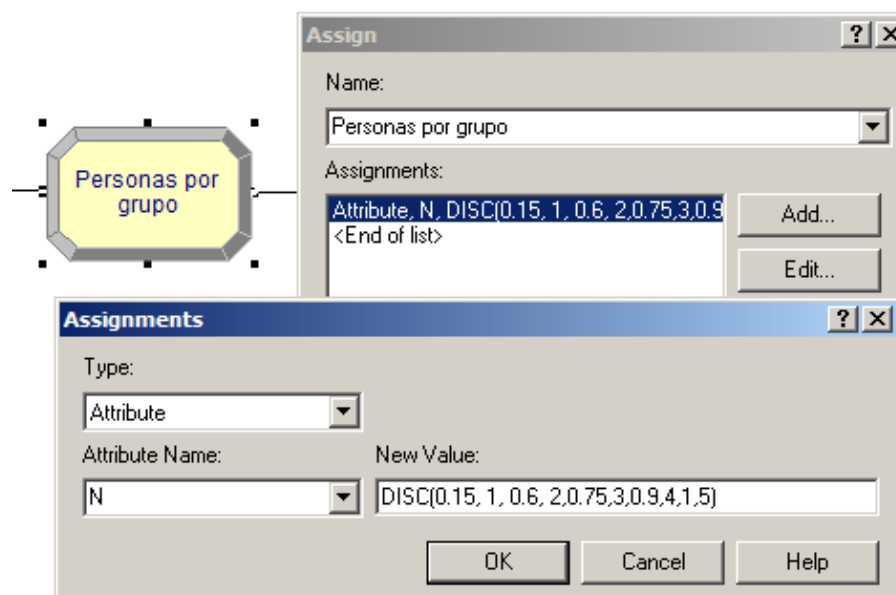


Figura 1.3: Módulo *Assign* en el cual se asigna a cada grupo un atributo que representa el número de personas que componen el grupo.

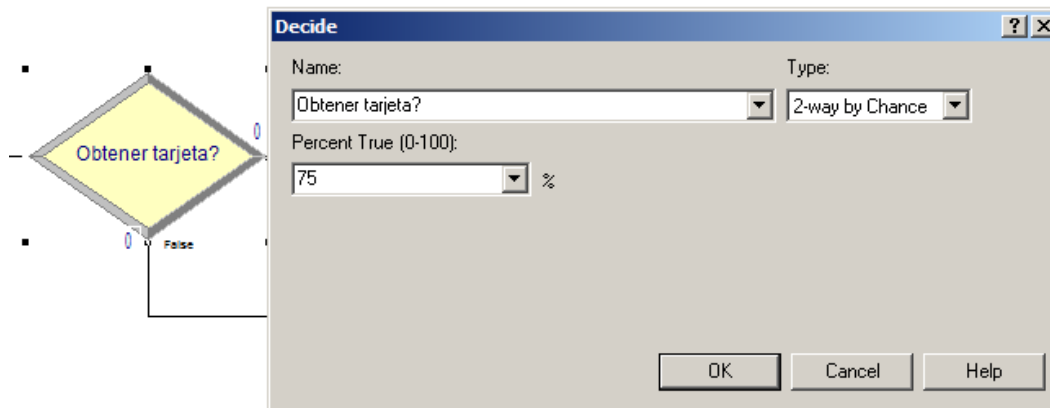


Figura 1.4: Módulo *Decide* que bifurca el flujo de los grupos en función de que dispongan o no de tarjeta de embarque.

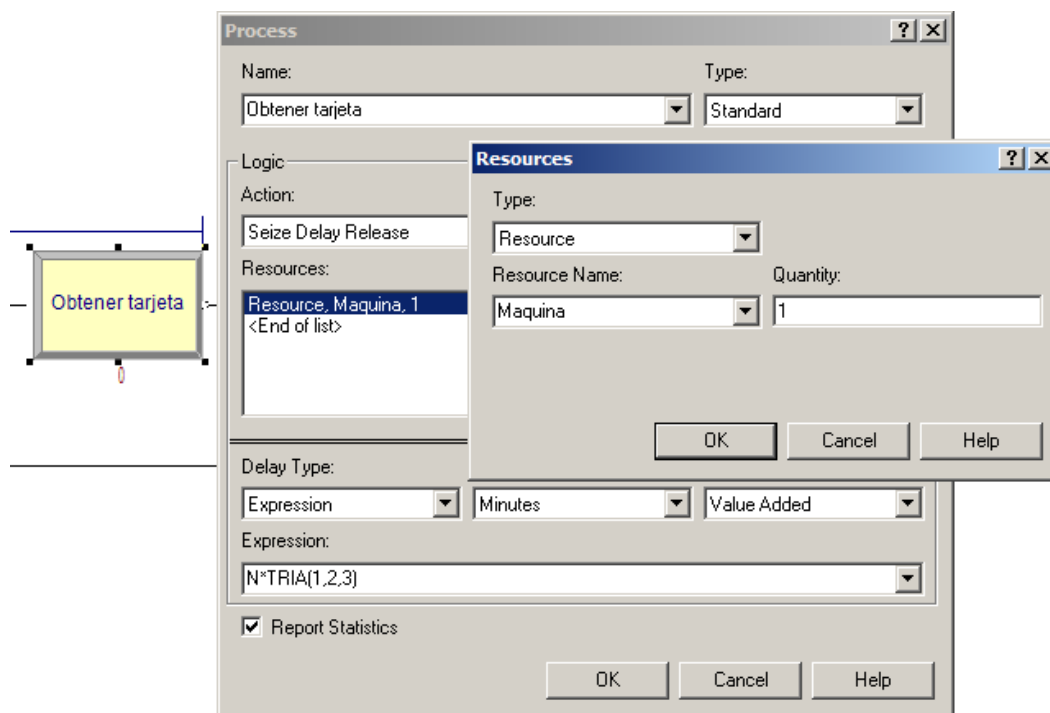


Figura 1.5: Proceso en el cual los pasajeros obtienen la tarjeta de embarque.

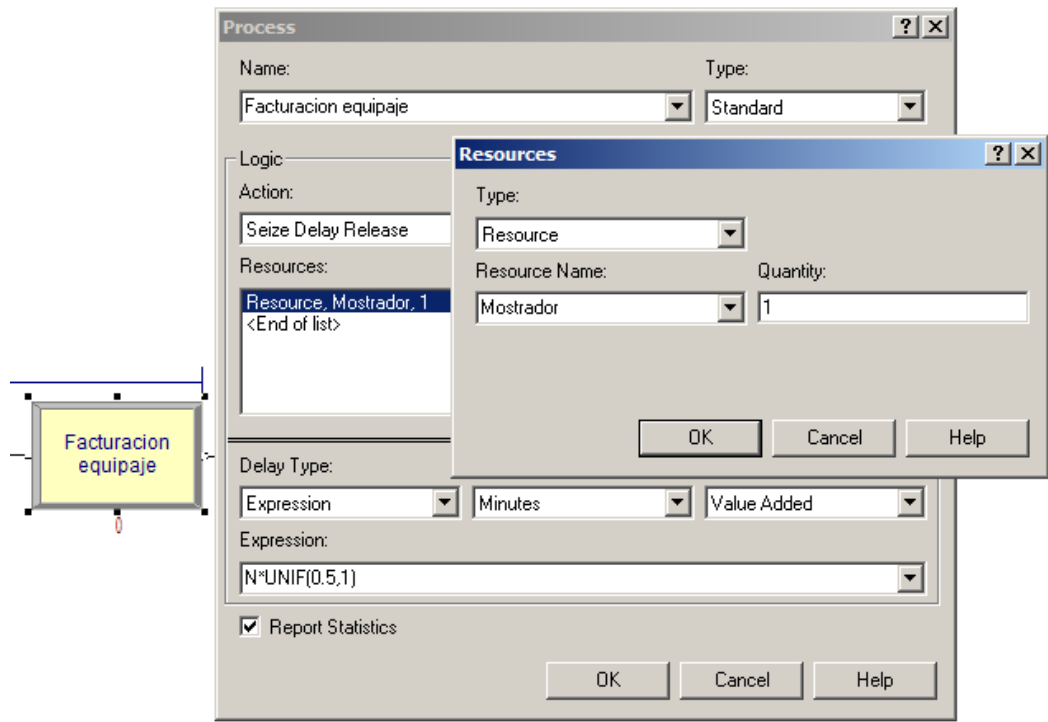


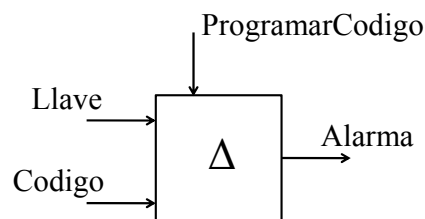
Figura 1.6: Proceso en el cual los pasajeros facturan su equipaje.

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	State Set Name	Failures	Report Statistics
1	Maquina	Fixed Capacity	4	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	Mostrador	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Figura 1.7: Módulo de datos *Resource*, en el cual se indica la capacidad de los recursos.

PREGUNTA 2 (3 puntos)

Describa, empleando el formalismo DEVS clásico, el comportamiento del sistema de alarma descrito a continuación. Como se muestra en la figura, el sistema tiene tres puertos de entrada (Llave,Codigo,ProgramarCodigo) y un puerto de salida (Alarma). Los eventos llegan a los puertos de entrada de uno en uno. Es decir, no se produce la llegada simultánea de varios eventos. Los eventos recibidos en el puerto Llave tienen valor 1. Los eventos recibidos en los puertos Codigo y ProgramarCodigo son números enteros en el rango $[0, 9999]$.



El sistema puede encontrarse en dos fases: “Pasivo” y “Activo”. Mientras el sistema está en la fase “Pasivo”, se comporta de la manera siguiente:

- Cuando se recibe un evento en el puerto ProgramarCodigo, el sistema almacena el valor del evento. Dicho valor almacenado se denomina Codigo-Seguridad.
- Se ignoran los eventos recibidos a través del puerto Codigo.
- Cuando se recibe un evento en el puerto Llave, el sistema pasa a la fase “Activo”.

Mientras el sistema está en la fase “Activo”, su comportamiento es el siguiente:

- Se ignoran los eventos recibidos a través de los puertos Llave y ProgramarCodigo.
- Cada Δ segundos se envía un evento de valor 1 a través del puerto Alarma. Es decir, en los instantes $\Delta, 2\Delta, 3\Delta, \dots$, donde Δ es un parámetro del modelo.

- Cuando se recibe un evento en el puerto *Codigo* y el valor de dicho evento coincide con *CodigoSeguridad*, entonces se envía un evento de valor 0 a través del puerto *Alarma* y el sistema pasa a la fase “Pasivo”. Si no coincide, se envía un evento de valor 2 a través del puerto *Alarma*, permaneciendo en este caso el sistema en la fase “Activo”.

Puede realizar todas las hipótesis adicionales que desee acerca del funcionamiento del sistema, siempre y cuando no estén en contradicción con las especificaciones anteriores.

Solución a la Pregunta 2

La descripción DEVS clásico del sistema se muestra en la Tabla 1.1. Obsérvese que el estado del sistema se define mediante las variables de estado siguientes: (*fase*, σ , *CodigoSeguridad*, *codAl*, *nD*). El significado de estas variables es el siguiente:

- La variable *fase* indica si el sistema se encuentra activo o pasivo. Esta variable tiene dos posibles valores: {“Pasivo”, “Activo”}.
- La variable σ almacena el tiempo que transcurrirá hasta la siguiente transición interna en ausencia de eventos externos. Puede tomar valores reales positivos, incluido el cero.
- La variable *CodigoSeguridad* almacena los eventos recibidos en el puerto *ProgramarCodigo* mientras el sistema está en la fase “Activo”. Puede tomar valores enteros en el intervalo [0, 9999].
- La variable *codAl* almacena el código de alarma correspondiente. Esta variable tiene los siguientes posibles valores: {0, 1, 2}.
- La variable *nD* almacena el tiempo restante hasta que el sistema envíe el siguiente evento con valor 1. Puede tomar valores reales positivos, incluido el cero.

Tabla 1.1: Modelo DEVS de la Pregunta 2.

Nombre del modelo	Ejercicio_2
Parámetros	$\Delta \in \mathbb{R}^+$;
X	(Llave, {1}), (Codigo, {0,1,...,9999}), (ProgramarCodigo, {0,1,...,9999})
Y	(Alarma, {0,1,2})
Variables de estado	$fase \in \{\text{"Pasivo"}, \text{"Activo"}\};$ $\sigma \in \mathbb{R}_{0,\infty}^+;$ $CodigoSeguridad \in \{0,1,\dots,9999\};$ $codAl \in \{0,1,2\};$ $nD \in \mathbb{R}_0^+;$
$\delta_{int} : S \rightarrow S$	if ($codAl == 0$) { return ($fase, \infty, CodigoSeguridad, codAl, \infty$); } if ($codAl == 1$) { return ($fase, \Delta, CodigoSeguridad, codAl, \Delta$); } if ($codAl == 2$) { return ($fase, nD, CodigoSeguridad, 1, nD$); }
$\delta_{ext} : Q \times X \rightarrow S$	if ($fase == \text{"Pasivo"}$ and $dato(ProgramarCodigo) \neq \emptyset$) { return ($fase, \infty, dato(ProgramarCodigo), codAl, \infty$); } if ($fase == \text{"Pasivo"}$ and $dato(Codigo) \neq \emptyset$) { return ($fase, \infty, CodigoSeguridad, codAl, \infty$); } if ($fase == \text{"Pasivo"}$ and $dato(Llave) \neq \emptyset$) { return ($\text{"Activo"}, \Delta, CodigoSeguridad, 1, \Delta$); } if ($fase == \text{"Activo"}$ and $dato(ProgramarCodigo) \neq \emptyset$) { return ($fase, \sigma - e, CodigoSeguridad, codAl, nD - e$); } if ($fase == \text{"Activo"}$ and $dato(Codigo) \neq \emptyset$) { if ($dato(Codigo) == CodigoSeguridad$); } { return ($\text{"Pasivo"}, 0, CodigoSeguridad, 0, \infty$); } else { return ($fase, 0, CodigoSeguridad, 2, nD - e$); } } if ($fase == \text{"Activo"}$ and $dato(Llave) \neq \emptyset$) { return ($fase, \sigma - e, CodigoSeguridad, codAl, nD - e$); }
$\lambda : S \rightarrow Y$	return (Alarma, $codAl$);
$ta : S \rightarrow \mathbb{R}_{0,\infty}^+$	return σ ;

PREGUNTA 3 (2 puntos)

Explique detalladamente cómo generar cada uno de los dos procesos de llegada siguientes. En cada caso, ilustre además sus explicaciones mediante un ejemplo en el cual genere varios instantes de llegada.

3.a (1 punto) Un proceso de Poisson estacionario.

3.b (1 punto) Un proceso de Poisson no estacionario.

Solución a la Pregunta 3

Véanse las Secciones 5.7.1 y 5.7.2 del texto base.

PREGUNTA 4 (2 puntos)

4.a (1 punto) Explique detalladamente qué es un generador de números pseudo-aleatorios de Tausworthe. Ponga un ejemplo.

4.b (1 punto) Explique detalladamente cómo emplear el generador de Tausworthe que ha puesto anteriormente como ejemplo para generar observaciones de una distribución uniforme con rango $[1, 2]$. Genere 5 observaciones, tomando para ello la semilla que desee.

Solución a la Pregunta 4

Véanse las Secciones 5.2.8 y 5.5.1 del texto base.