

## MODELADO Y SIMULACIÓN

Febrero de 2019 - Segunda semana

### PREGUNTA 1 (3 puntos)

Se pretende estudiar mediante simulación una carnicería que vende dos tipos de productos: “al corte” y “envasados”. Los productos envasados se encuentran colocados en una estantería accesible al público, de modo que los clientes pueden cogerlos sin necesidad de esperar cola. Los productos al corte son despachados en un mostrador por un empleado, de modo que si éste se encuentra ocupado atendiendo a un cliente, los demás clientes deben esperar en una cola con disciplina FIFO.

La carnicería está atendida por dos empleados. El primero se dedica exclusivamente a despachar los productos al corte. El segundo se dedica exclusivamente a cobrar en la caja.

Se supone que los clientes llegan a la carnicería de uno en uno. El intervalo de tiempo entre llegadas sucesivas de clientes está distribuido exponencialmente, con media 5 minutos.

Cada cliente compra un cierto número aleatorio de productos envasados y un cierto número aleatorio de productos al corte. La distribución de probabilidad del número de productos envasados que compra un cliente es la siguiente:

1 producto	–	probabilidad: 0.20
2 productos	–	probabilidad: 0.15
3 productos	–	probabilidad: 0.35
4 productos	–	probabilidad: 0.20
5 productos	–	probabilidad: 0.10

La distribución de probabilidad del número de productos al corte que compra un cliente es:

0 productos	–	probabilidad: 0.20
1 producto	–	probabilidad: 0.35
2 productos	–	probabilidad: 0.35
3 productos	–	probabilidad: 0.10

Se supone que el tiempo que tarda el cliente en coger cada producto envasado es igual a 1 minuto. Por ejemplo, un cliente que compre dos productos envasados, tarda dos minutos en cogerlos. Asimismo, se supone que el tiempo que tarda el primer empleado en despachar cada producto al corte es igual a 3 minutos. Por ejemplo, este empleado tarda 6 minutos en atender a un cliente que compre dos productos al corte.

El funcionamiento del sistema es el siguiente. Cuando un cliente llega a la carnicería, en primer lugar coge los productos envasados que necesita. A continuación, si el número de productos al corte que necesita es cero, se dirige a la caja. En caso contrario (es decir, si va a comprar productos al corte), se dirige al mostrador atendido por el primer empleado. Si éste se encuentra libre, atiende al cliente. Si se encuentra ocupado, el cliente se pone a la cola y espera su turno. Una vez el cliente ha sido atendido, se dirige a la caja para pagar.

La carnicería tiene una única caja de cobro, que es atendida por el segundo empleado. Si cuando llega el cliente a la caja el segundo empleado se encuentra libre, entonces el cliente es atendido. Si el segundo empleado se encuentra ocupado, el cliente espera en una cola con disciplina FIFO. El tiempo necesario para atender en la caja a un cliente es la suma de dos tiempos:

- El tiempo que tarda el empleado en leer el código de barras de todos los productos, que es igual a 30 segundos para cada producto.
- El tiempo de cobro, que se supone que es 1 minuto para cada cliente.

Por ejemplo, si un cliente ha comprado 1 producto envasado y 3 productos al corte, el segundo empleado tardará  $0.5 \times 4 + 1 = 3$  minutos en cobrarle. Una vez ha pagado, el cliente abandona la carnicería.

El horario de apertura al público de la carnicería es de 10:00h a 14:00h y de 16:00h a 20:00h. No obstante, al realizar el modelo se hace la simplificación de suponer que la carnicería abre cada día 8 horas ininterrumpidamente, y que una vez alcanzada la hora de cierre los clientes que están en la carnicería la abandonan.

Describa *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. El objetivo del estudio es estimar:

- La utilización de cada uno de los dos empleados.
- El tiempo medio de espera en cola del mostrador de productos al corte.
- El tiempo medio de espera en cola de la caja.

### **Solución a la Pregunta 1**

En la Figura 1.1 se muestra el diagrama de módulos del sistema y en la Figura 1.2 el proceso de llegada de entidades. Cada entidad representa un cliente.

Se asocian dos atributos a cada entidad: el número de productos envasados y el número de productos al corte que compra dicho cliente. Dicha asignación se realiza en un módulo Assign, tal como se muestra en la Figura 1.3.

El cliente coge en primer lugar los productos envasados. Este proceso está representado en la Figura 1.4. La acción es de tipo delay.

Mediante un módulo de decisión, representado en la Figura 1.5, se bifurca el flujo de entidades, dependiendo de que vayan o no a comprar productos al corte. La definición de los procesos de compra de productos al corte y de pago en caja se muestra en las Figuras 1.6 y 1.7, respectivamente. En la Figura 1.8 se muestran los recursos del sistema: el empleado que realiza los productos al corte (empleado1) y el que trabaja en la caja (empleado2).

Finalmente, en la Figura 1.9 se muestra la definición del experimento.

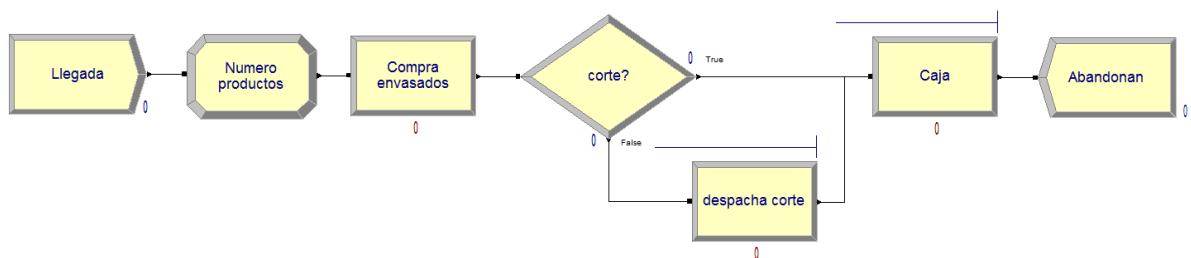


Figura 1.1: Diagrama de módulos del sistema.

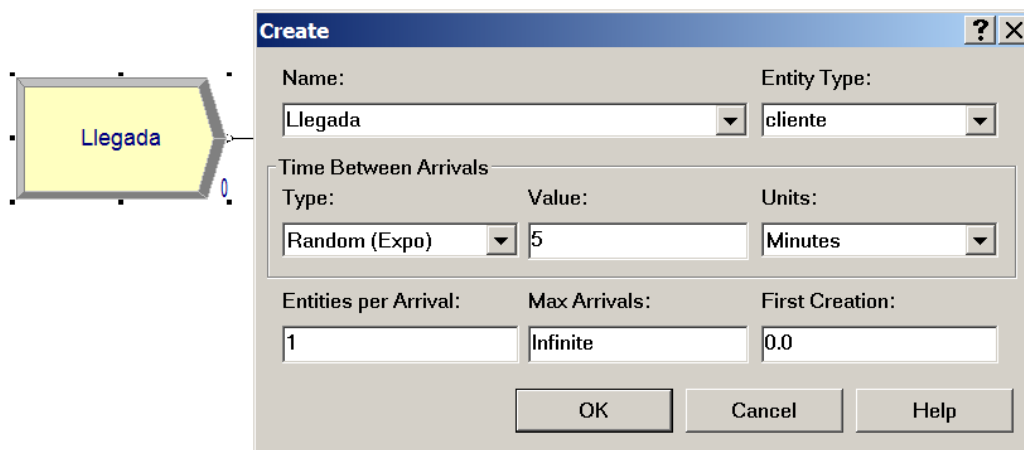


Figura 1.2: Proceso de llegada de clientes.

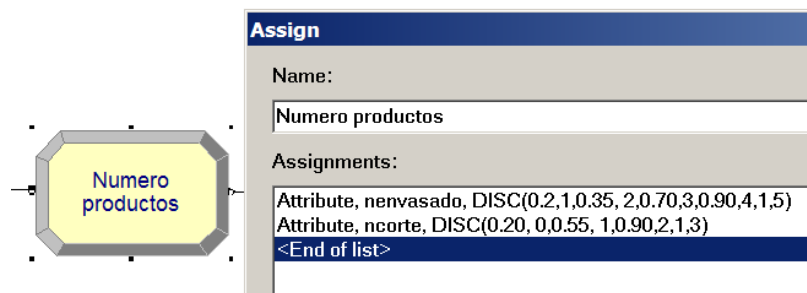


Figura 1.3: Asignación de atributos a las entidades.

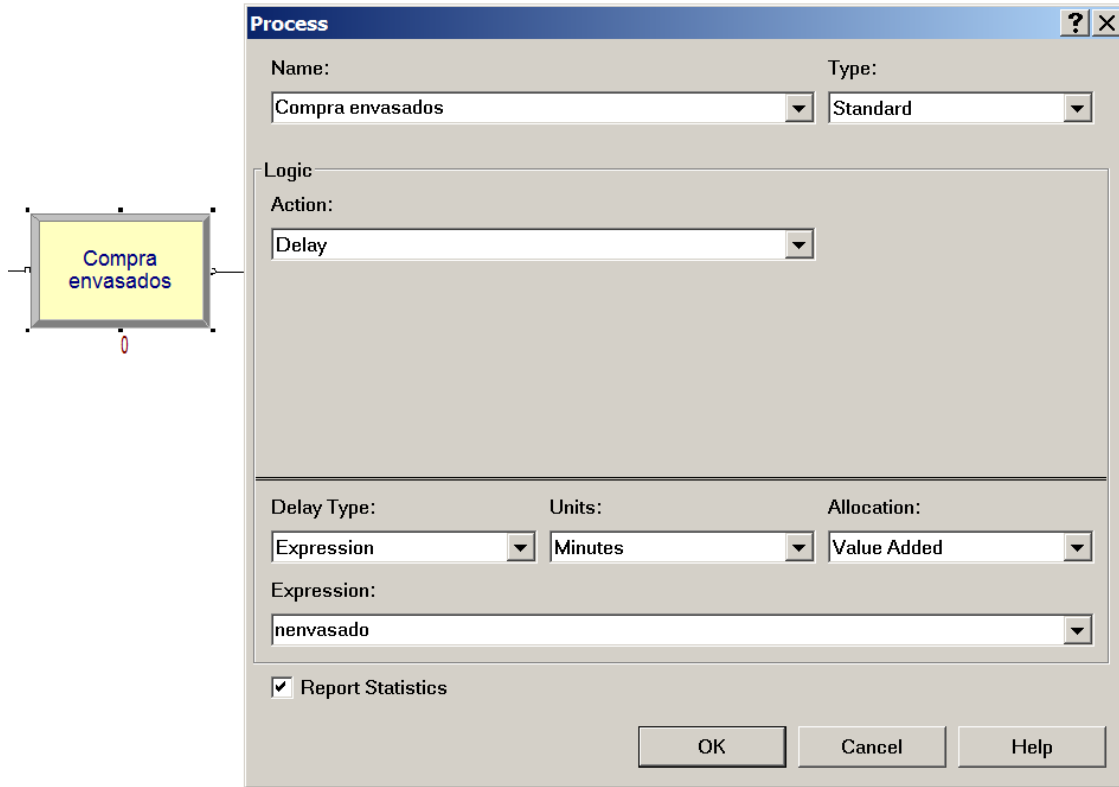


Figura 1.4: El cliente coge los productos envasados.

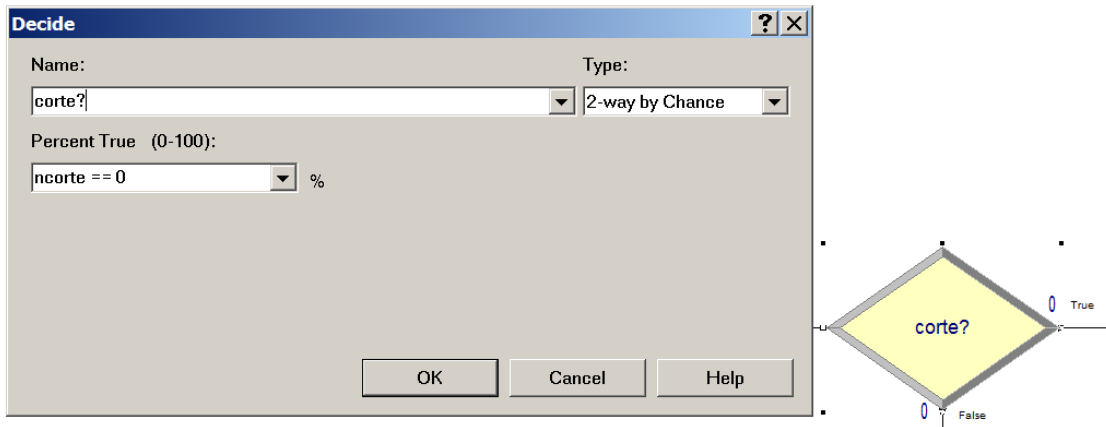


Figura 1.5: ¿Va el cliente a no comprar productos al corte?

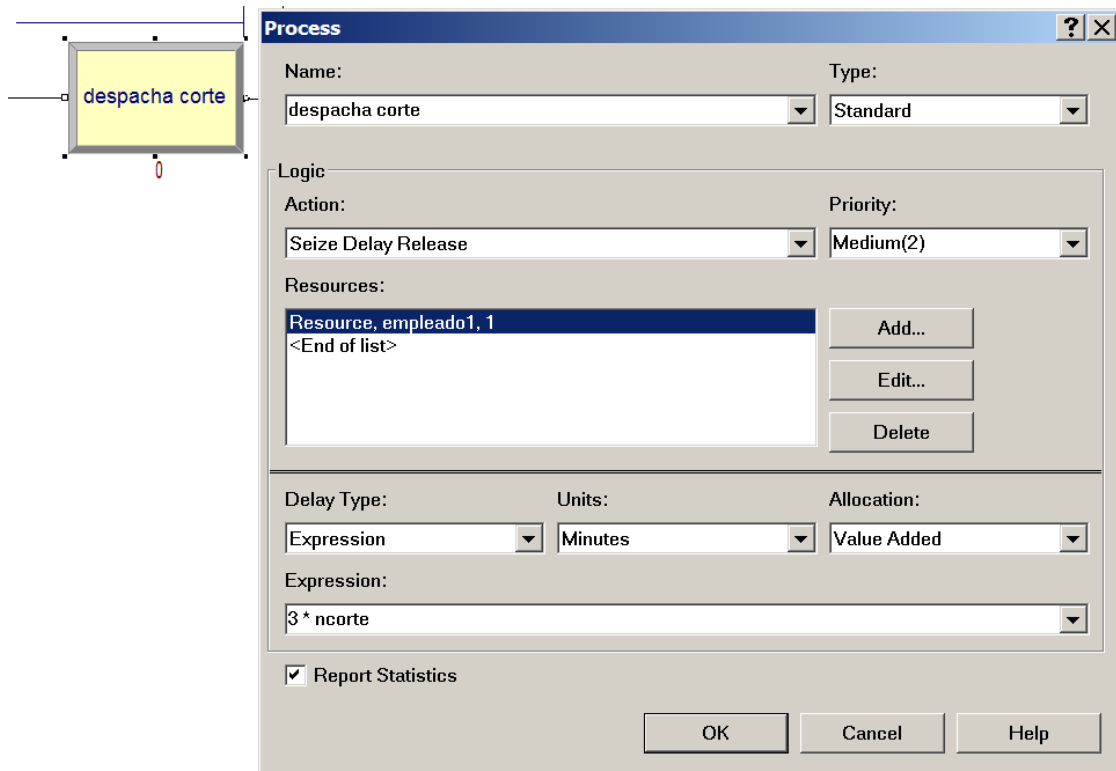


Figura 1.6: Proceso de compra de productos al corte.

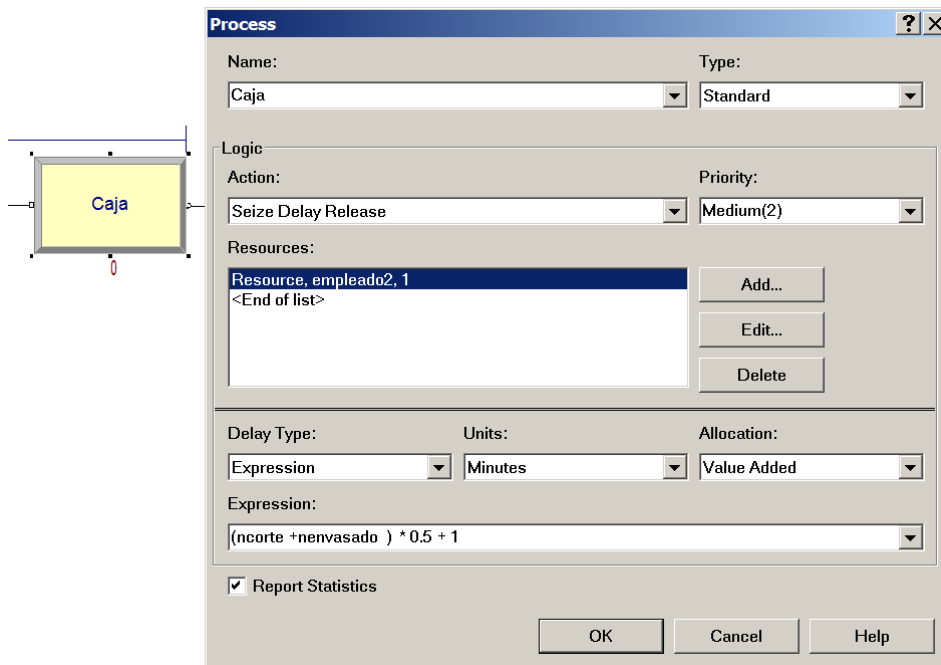


Figura 1.7: Pago en la caja.

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	empleado1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	empleado2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Double-click here to add a new row.

Figura 1.8: Definición de los recursos.

**Run Setup**

Run Speed | Run Control | Reports | Project Parameters  
 Replication Parameters | Array Sizes | Arena Visual Designer

Number of Replications:

Initialize Between Replications  
 Statistics     System

Start Date and Time:

Warm-up Period:     Time Units:

Replication Length:     Time Units:

Hours Per Day:

Base Time Units:

Terminating Condition:

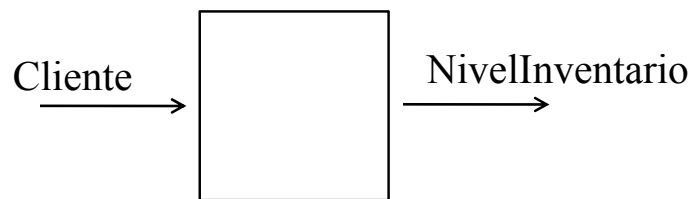
Aceptar    Cancelar    Aplicar    Ayuda

Figura 1.9: Definición del experimento.

**PREGUNTA 2** (3 puntos)

Describa, empleando el formalismo DEVS clásico, el modelo descrito a continuación del inventario de una estantería que contiene productos envasados.

La interfaz del modelo, que se muestra en la figura, está compuesta por un puerto de entrada (Cliente) y un puerto de salida (NivelInventario).



Los eventos de entrada tienen valor “cliente”. La recepción de un evento en el puerto de entrada indica la llegada de un cliente a la estantería, dispuesto a llevarse de la misma un número de productos que es una observación independiente de la distribución de probabilidad siguiente:

- 1 producto – probabilidad: 0.20
- 2 productos – probabilidad: 0.15
- 3 productos – probabilidad: 0.35
- 4 productos – probabilidad: 0.20
- 5 productos – probabilidad: 0.10

Supongamos que el nivel de inventario (número de productos envasados que hay en la estantería) antes de la llegada del cliente es  $N_{antes}$  y que el cliente desea llevarse  $M$  productos.

- Si  $N_{antes} \geq M$ , el cliente puede llevarse los productos que desea y el nuevo valor del inventario es  $N_{despues} = N_{antes} - M$ .
- Si  $N_{antes} < M$ , el cliente sólo puede llevarse  $N_{antes}$  productos, siendo el nuevo valor del inventario  $N_{despues} = 0$ .

En el instante en que se recibe el evento de entrada, el modelo calcula el nuevo valor del inventario y transmite dicho valor a través del puerto de salida.

Puede realizar todas las hipótesis adicionales que desee acerca del funcionamiento del modelo, 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 puede definirse mediante las variables de estado siguientes:  $(fase, \sigma, N)$ .

- La variable  $fase$  puede tomar los valores {"pasivo", "activo"}.
- La variable  $\sigma$  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  $N$  almacena el nivel del inventario, es decir, el número de productos envasados que hay en la estantería.

Supongamos que la estantería contiene inicialmente 100 productos. El estado inicial del sistema es: ("pasivo",  $\infty$ , 100).

**Tabla 1.1:** Modelo DEVS de la Pregunta 2.

Nombre del modelo	Ejercicio_2
Parámetros	
$X$	(Cliente, "cliente")
$Y$	(NivelInventario, $\mathbb{N} \cup \{0\}$ )
Variables de estado	$fase \in \{\text{"pasivo"}, \text{"activo"}\};$ $\sigma \in \mathbb{R}_{0,\infty}^+;$ $N \in \mathbb{N} \cup \{0\};$
$\delta_{int} : S \rightarrow S$	<b>return</b> ("pasivo", $\infty$ , $N$ );
$\delta_{ext} : Q \times X \rightarrow S$	$M \in \{1, 2, 3, 4, 5\};$ $M = \text{DISC}(0.2, 1, 0.35, 2, 0.70, 3, 0.90, 4, 1, 5);$ <b>return</b> ("activo", 0, $\text{máx}(N - M, 0)$ );
$\lambda : S \rightarrow Y$	<b>return</b> (NivelInventario, $N$ );
$ta : S \rightarrow \mathbb{R}_{0,\infty}^+$	<b>return</b> $\sigma$ ;

**PREGUNTA 3** (2 puntos)

Describa *detalladamente* un algoritmo para generar observaciones del número de productos envasados que compra cada cliente. Emplee el *método de la transformación inversa*.

**Solución a la Pregunta 3**

Véase la Sección 5.4.1 del texto base.

**PREGUNTA 4** (2 puntos)

- 4.a** (1 punto) Explique en qué consiste un *diseño experimental factorial completo*. Ponga un ejemplo, basándose en el sistema descrito en la Pregunta 1, que tenga 3 factores y 2 niveles por factor.
- 4.b** (1 punto) Explique en qué consiste un *diseño experimental factorial fraccional*. Ponga un ejemplo de diseño factorial fraccional  $2^{3-1}$  basándose en el sistema descrito en la Pregunta 1.

**Solución a la Pregunta 4**

Véanse las Secciones 7.3 y 7.4 del texto base.