

MODELADO Y SIMULACIÓN

Solución al Ejercicio de Autocomprobación 2

PREGUNTA 1 (3 puntos)

Se pretende estudiar mediante simulación el funcionamiento de una cadena de montaje dedicada al ensamblado, prueba y empaquetado de circuitos electrónicos. El funcionamiento del sistema se describe a continuación.

Al sistema llegan solicitudes de fabricación de un cierto número de unidades del circuito. El tiempo que transcurre entre la llegada de dos solicitudes consecutivas está distribuido exponencialmente, con media 1 hora. El número de unidades del circuito que se solicita fabricar en cada solicitud está distribuido de la forma siguiente:

Circuitos a fabricar	Probabilidad
10	0.25
20	0.25
50	0.40
100	0.10

Las solicitudes de fabricación esperan en una única cola FIFO frente a las máquinas ensambladoras. Cada circuito es ensamblado, probado y empaquetado independientemente de los demás.

Cada máquina ensambladora ensambla de principio a fin el circuito. Se estima que el tiempo necesario para el ensamblaje del circuito está distribuido triangularmente, con rango $[1, 6]$ minutos y media 3 minutos. El sistema dispone de 10 máquinas ensambladoras, que funcionan independientemente entre sí.

Los circuitos ensamblados se ponen en la cola FIFO de prueba. Dicha cola es atendida por 5 máquinas de prueba, que funcionan independientemente entre sí.

El tiempo necesario para probar un circuito está distribuido de forma normal, con media 180 segundos y desviación estándar 30 segundos.

La probabilidad de que el circuito supere la prueba es del 90 %. Los circuitos que superan la prueba son puestos en la cola del proceso de empaquetado. Los que no superan la prueba son puestos en la cola FIFO del proceso de reparación.

El proceso de reparación es realizado por dos operarios, que trabajan independientemente entre sí. Se estima que el tiempo que necesita el operario para decidir si el circuito puede repararse y para, en su caso, repararlo está distribuido de forma normal, con media 15 minutos y desviación estándar 3 minutos.

Sólo el 50 % de los circuitos que han fallado la prueba pueden ser reparados. Los que no pueden ser reparados son desechados. Los que han sido reparados son puestos en cola del proceso de empaquetado.

El proceso de empaquetado es realizado por las máquinas empaquetadoras. Hay 3 máquinas empaquetadoras, que funcionan independientemente entre sí. Se estima que el tiempo necesario para que una máquina empaquetadora empaquete un circuito está distribuido uniformemente, con rango [1, 3] minutos. Los circuitos empaquetados abandonan el sistema.

El objetivo del estudio es estimar la utilización de los recursos. Para ello, se realiza una simulación en el estacionario. Se simula el funcionamiento del sistema durante 1000 horas.

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.

Solución a la Pregunta 1

En la Figura 1.1 se muestra una posible manera de describir el funcionamiento del sistema. El proceso de llegada de solicitudes puede ser descrito tal como se muestra en la Figura 1.2. La llegada de una solicitud se interpreta como la llegada de cierto número de entidades “circuito” que van a ser fabricadas.

Los procesos “Ensamblado”, “Prueba”, “Empaquetado” y “Reparacion” son del tipo *Seize-Delay-Release*. En las Figuras 1.3 y 1.4 se muestra la especificación de dichos procesos. En la Figura 1.5 se muestra el contenido de los módulos de datos *Process* y *Resource*. El contenido del módulo de datos *Process* se completa automáticamente al ir definiendo los procesos, tal como se muestra en las Figu-

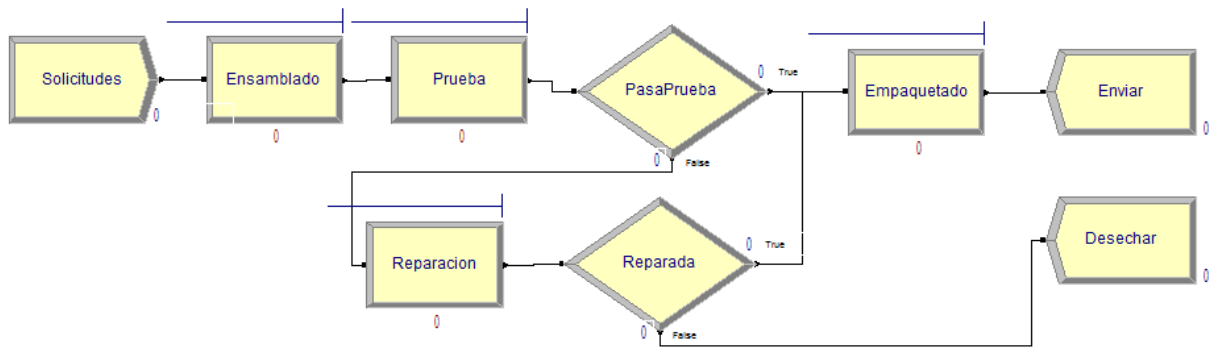


Figura 1.1: Diagrama de módulos.

Create - Basic Process								
	Name	Entity Type	Type	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Solicitudes	circuito	Expression	EXPO(1)	Hours	DISC(0.25, 10, 0.5, 20, 0.90,50,1,100)	Infinite	0.0

Figura 1.2: Módulo *Create* que describe la llegada de solicitudes de fabricación de circuitos.

ras 1.3 y 1.4. Por el contrario, la capacidad de los recursos debe especificarse en el bloque de datos *Resource*.

Los módulos *Decide* “PasaPrueba” y “Reparada” son del tipo *two-way by chance* (véase la Figura 1.6).

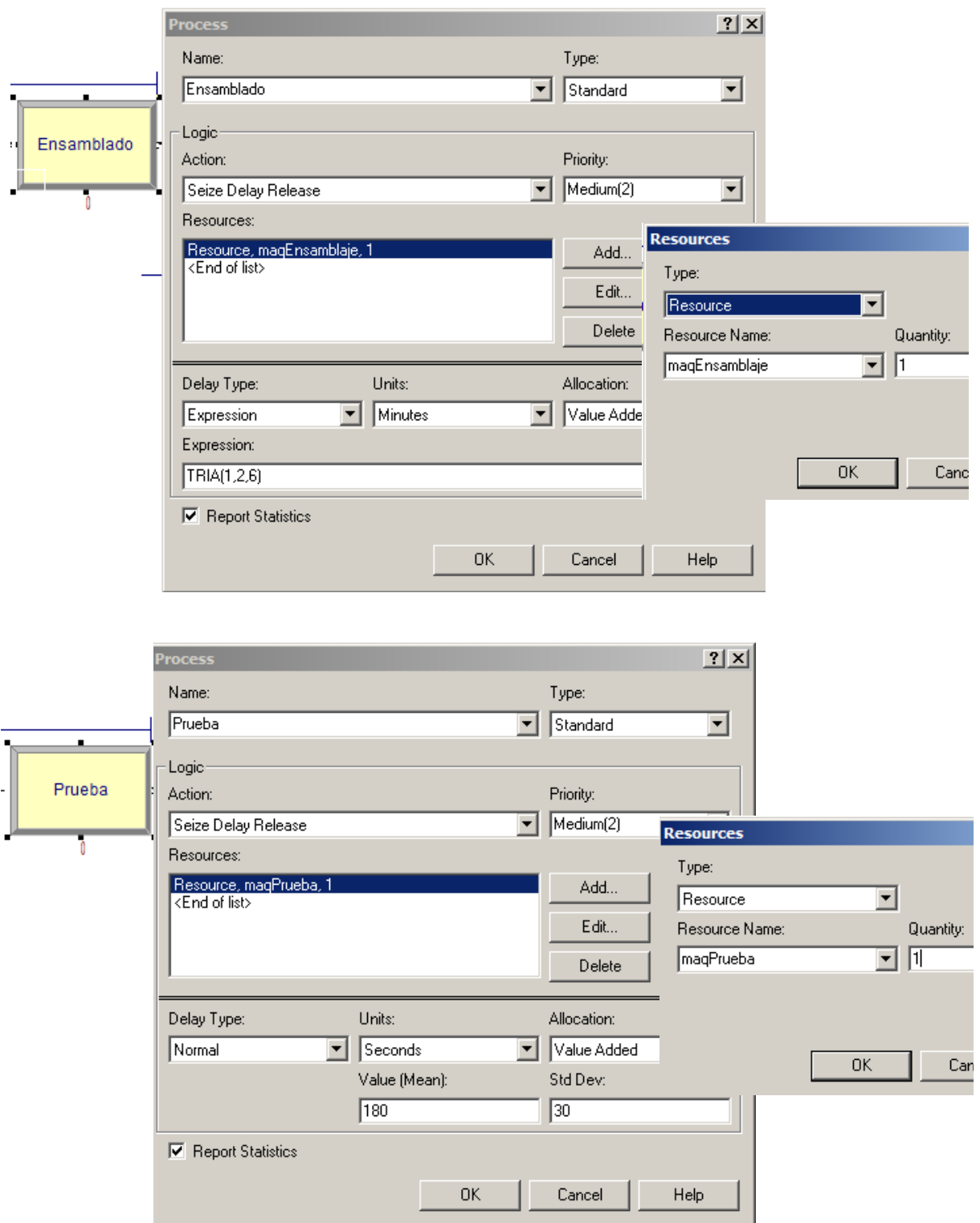


Figura 1.3: Procesos de ensamblado y prueba.

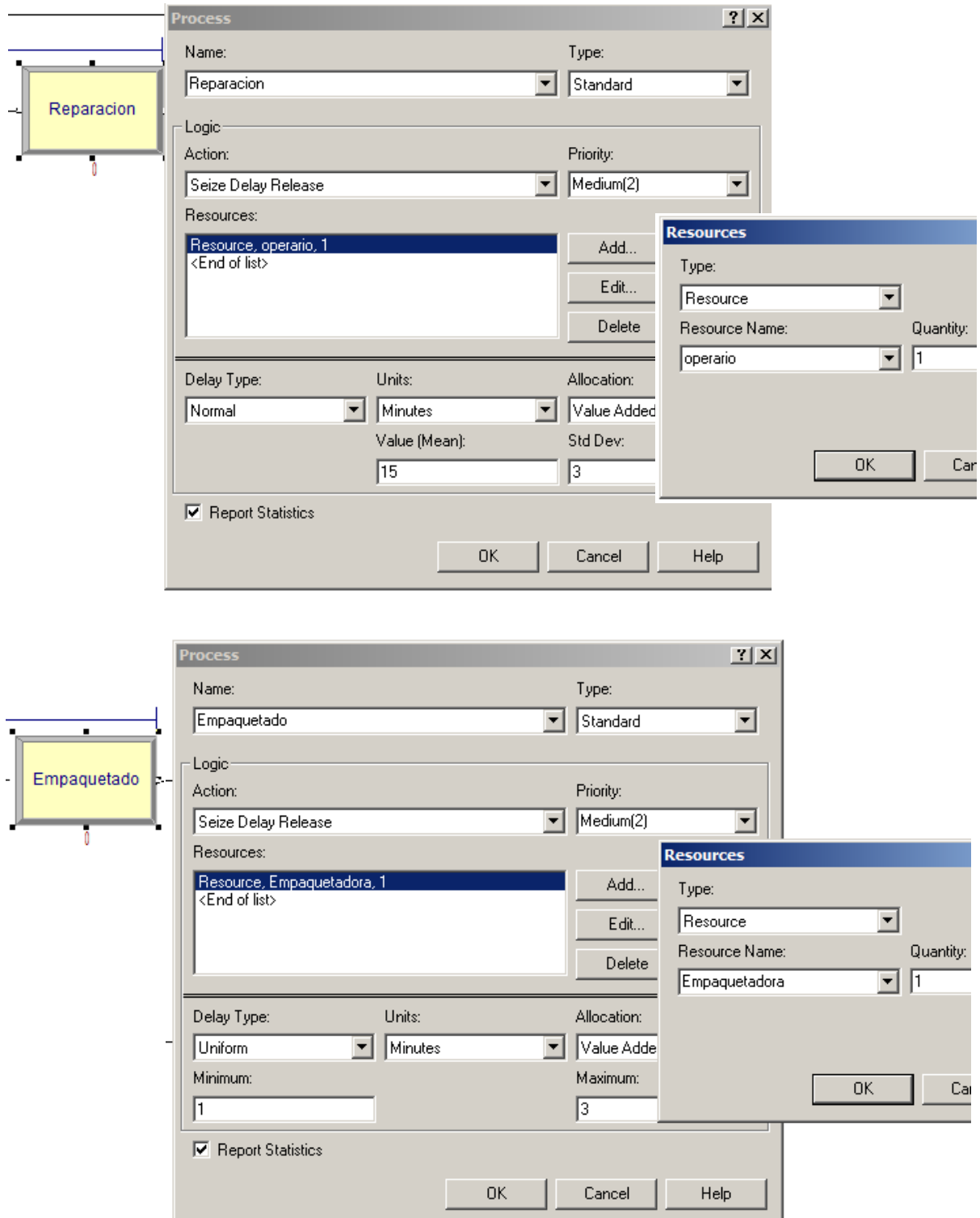


Figura 1.4: Procesos de reparación y empaquetado.

Process - Basic Process													
	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Std Dev	Expression
1	Ensamblado	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added		2	5	2	TRIA(1,2,6)
2	Prueba	Standard	Seize Delay Release	Medium(2)	1 rows	Normal	Seconds	Value Added	5	180	1.5	30	1
3	Reparacion	Standard	Seize Delay Release	Medium(2)	1 rows	Normal	Minutes	Value Added	5	15	1.5	3	1
4	Empaquetado	Standard	Seize Delay Release	Medium(2)	1 rows	Uniform	Minutes	Value Added	1	1	3	2	1

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	State Set Name	Failures	Report Statistics
1	maqEnsamblaje	Fixed Capacity	10	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	maqPrueba	Fixed Capacity	5	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	Empaquetadora	Fixed Capacity	3	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
4	operario	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

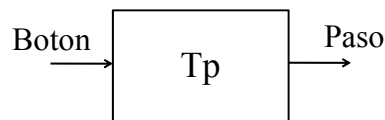
Figura 1.5: Resumen de los procesos y definición de la capacidad de los recursos.

The figure shows two 'Decide' dialog boxes and their graphical representation in a flowchart. The top dialog box is for a decision named 'PasaPrueba' with a '2-way by Chance' type and a '90%' probability of being true. The bottom dialog box is for a decision named 'Reparada' with a '2-way by Chance' type and a '50%' probability of being true. The flowchart on the left shows two diamond-shaped decision nodes: 'PasaPrueba' and 'Reparada', each with 'True' and 'False' paths leading to subsequent process steps.

Figura 1.6: Definición de los módulos de decisión.

PREGUNTA 2 (3 puntos)

Empleando el formalismo DEVS clásico, describa el modelo del controlador de un semáforo descrito a continuación. Como se muestra en la figura, el modelo tiene un puerto de entrada (Boton) y un puerto de salida (Paso). El modelo tiene un parámetro, T_p , que puede tomar valores reales positivos.



Los eventos llegan al puerto de entrada de uno en uno. Los eventos recibidos en el puerto Boton tienen valor {"pulsa"}.

El sistema puede encontrarse en dos fases: {"pasanPeatones", "pasanCoches"}. Inicialmente el sistema está en la fase "pasanCoches".

Cuando el sistema está en la fase "pasanCoches" y recibe un evento en el puerto Boton:

1. El sistema envía un evento de valor "turnoPeatones" a través del puerto Paso y pasa a la fase "pasanPeatones".
2. Durante los siguientes T_p segundos se ignoran los eventos recibidos en el puerto Boton.
3. Transcurridos T_p segundos, se envía un evento de valor "turnoCoches" a través del puerto Paso y el sistema pasa a la fase "pasanCoches", quedando el sistema listo para volver a responder a la llegada de un evento al puerto Boton.

El sistema ignora los eventos recibidos en el puerto Boton mientras está en la fase "pasanPeatones".

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 siguiente.

Nombre del modelo	DEVS_semaforo
Parámetros	$T_p \in \mathbb{R}^+$;
X	(Boton, {"pulsa"})
Y	(Paso, {"turnoPeatones", "turnoCoches"})
Variables de estado	fase \in {"pasanPeatones", "pasanCoches"}; $\sigma \in \mathbb{R}_{0,\infty}^+$;
$\delta_{int} : S \rightarrow S$	if (fase == "pasanCoches") { return ("pasanPeatones", T_p); } else { return ("pasanCoches", ∞); }
$\delta_{ext} : Q \times X \rightarrow S$	if (fase == "pasanCoches") { return (fase, 0); } else { return (fase, $\sigma - e$); }
$\lambda : S \rightarrow Y$	if (fase == "pasanCoches") { return (Paso, "turnoPeatones"); } else { return (Paso, "turnoCoches"); }
$ta : S \rightarrow \mathbb{R}_{0,\infty}^+$	return σ ;

El valor inicial de las variables de estado es ("pasanCoches", ∞).

PREGUNTA 3 (2 puntos)

Describa detalladamente un algoritmo para generar observaciones de las dos distribuciones siguientes empleando el método de la transformación inversa:

- 3.a) (1 punto) Distribución exponencial con media 15 minutos.
- 3.b) (1 punto) Una distribución discreta de probabilidad cuyos valores y probabilidades son:

Valor	Probabilidad
1	0.6
2	0.2
3	0.15
4	0.05

Solución a la Pregunta 3

Véase la Sección 5.4.1, y los Ejercicios 5.6 y 5.7 del texto base.

PREGUNTA 4 (2 puntos)

- 4.a) (1 punto) Explique detalladamente cuál es la finalidad del método gráfico de Welch.
- 4.b) (1 punto) Explique detalladamente los pasos de que consta el método gráfico de Welch.

Solución a la Pregunta 4

Véase la Sección 6.4.1 del texto base.