

ÍNDICE

1. - CUESTIONES GENERALES SOBRE SIMULACIÓN:

- * **Introducción**
- * **Sistemas y modelos**
- * **Simulación de modelos de sucesos discretos**
- * **Componentes y organización de un estudio de simulación**
- * **Ejemplo de simulación de una cola con un servidor**
- * **Etapas más frecuentes en un estudio de simulación**
- * **Ventajas, inconvenientes y errores más frecuentes en los estudios de simulación**

ÍNDICE

2.-GENERACIÓN DE NÚMEROS ALEATORIOS.

3.-DECIDIR EN UNA SIMULACIÓN CUÁNDO SE ALCANZA EL ESTADO ESTACIONARIO.

4.- ANÁLISIS ESTADÍSTICO DE LOS RESULTADOS:

a) Estimación de medidas de rendimiento en estado estacionario.


b) Intervalos de confianza.

c) Número de repeticiones de la simulación (réplicas) y longitud de las mismas.

d) Obtención de réplicas independientes.

5.-INDICACIONES PARA REALIZAR EL TRABAJO DE SIMULACIÓN.

1. INTRODUCCIÓN

 **SIMULACIÓN:** Técnicas para imitar el funcionamiento de sistemas o procesos reales mediante programas de ordenador.

 **ÁREAS DE APLICACIÓN:**

- Diseño y análisis de sistemas de producción
- Evaluación de requerimientos hardware y software para un sistema de computación
- Diseño de sistemas de comunicación
- Diseño y operación de sistemas de transporte como aeropuertos, puertos o metro
- Evaluación de diseños para la organización de servicios como hospitales u oficinas de correos
- Análisis de sistemas económicos o financieros

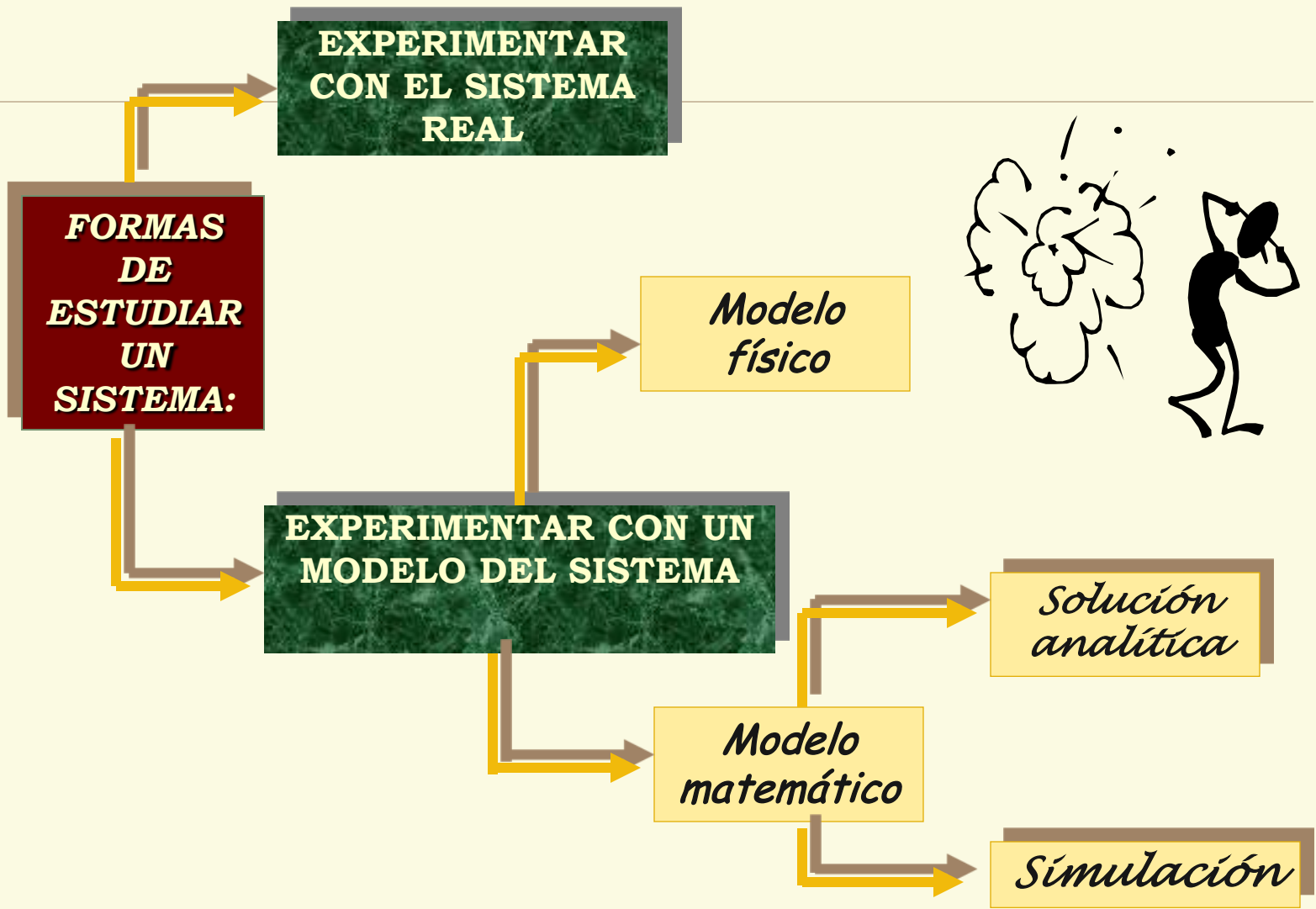
2. SISTEMAS Y MODELOS

SISTEMA: Colección de entidades que interactúan entre sí para conseguir un determinado fin.

MODELO: Representación simplificada de un sistema que se utiliza para estudiar su comportamiento.

HIPÓTESIS SOBRE EL MODELO:

- **Sencillas:** posible obtener soluciones exactas o analíticas .
- **Complejas:** es lo más usual. Deben ser estudiados mediante simulación. Se obtienen soluciones aproximadas.



FORMAS DE ESTUDIAR UN SISTEMA:

EXPERIMENTAR CON EL SISTEMA REAL

Modelo físico

EXPERIMENTAR CON UN MODELO DEL SISTEMA

Modelo matemático

Solución analítica

Simulación



MODELOS DE SIMULACIÓN

- Estudiaremos un sistema mediante un **MODELO MATEMÁTICO** cuyas soluciones se obtienen mediante **SIMULACIÓN**.
- Este tipo de modelos se llaman **MODELOS DE SIMULACIÓN**.
- El **ESTADO DEL SISTEMA** se describe mediante las **VARIABLES DE ESTADO**: colección de variables necesarias para describir el sistema en un instante dado.

TIPOS DE MODELOS DE SIMULACIÓN

- Clasificación de los modelos de simulación:
 - Modelos estáticos frente a modelos dinámicos.
 - Modelos determinísticos frente a modelos estocásticos.
 - Modelos discretos frente a modelos continuos.
- Estudiaremos modelos de simulación **DINÁMICOS, ESTOCÁSTICOS Y DISCRETOS**

3. SIMULACIÓN DE MODELOS DE SUCEOS DISCRETOS

SUCESO: ocurrencia que puede modificar el estado del sistema

EJEMPLO: Sistema de colas con un servidor

📄 **SUCESOS:** Llegadas de clientes, salidas de clientes servidos.

📄 **OBJETIVO:** Estimar tiempo medio en sistema de un cliente.

📄 **VARIABLES DE ESTADO:** Estado del servidor, instante de llegada de cada cliente.

Observación: si el objetivo fuese diferente, las variables de estado también lo serían.

4. COMPONENTES Y ORGANIZACIÓN DE UN ESTUDIO DE SIMULACION

Aunque la simulación se usa en campos diversos, existen una serie de componentes comunes:

➤ **ESTADO DEL SISTEMA:** colección de variables (variables de estado) necesarias para describir el sistema en un instante dado.

➤ **RELOJ DE SIMULACION :** variable que proporciona el instante actual de tiempo simulado. Se actualiza cuando ocurre un suceso.

4. COMPONENTES Y ORGANIZACIÓN DE UN ESTUDIO DE SIMULACION

- **LISTA DE SUCESOS:** la lista que contiene los próximos instantes en los que van a ocurrir los distintos tipos de sucesos (en un sistema de colas con un servidor, por ejemplo, contiene dos valores).
- **CONTADORES ESTADISTICOS:** Variables utilizadas para guardar la información estadística sobre el funcionamiento del sistema (depende de las medidas de rendimiento que se quieran estimar).

4. COMPONENTES Y ORGANIZACIÓN DE UN ESTUDIO DE SIMULACION

- *SUBROUTINA DE INICIALIZACION:* subprograma que inicializa el modelo de simulación en el instante cero.
- *SUBROUTINA DE TIEMPOS:* Subprograma que determina el siguiente suceso de la lista de sucesos, y avanza el reloj hasta el instante en que ese suceso ocurre.
- *SUBROUTINAS DE SUCECOS:* Subprograma que actualiza las variables de estado y los contadores cuando ocurre un suceso (una rutina para cada tipo de suceso)

4. COMPONENTES Y ORGANIZACIÓN DE UN ESTUDIO DE SIMULACION

- *LIBRERIA DE RUTINAS*: conjunto de subprogramas que generan números aleatorios (valores independientes de una distribución) de las distribuciones utilizadas en el modelo.
- *GENERADOR DE INFORMES*: Subprograma que estima las medidas de comportamiento del sistema y da un informe con los resultados de la simulación.
- *PROGRAMA PRINCIPAL*: Llama a la rutina de tiempos para determinar el próximo suceso y transfiere el control a la rutina correspondiente . Verifica la condición de parada y llama al generador de informes cuando la simulación ha terminado.

RUTINA DE INICIALIZACIÓN

- Pone el reloj a cero
- Inicializa el estado del sistema y los contadores
- Inicializa la lista de sucesos

PROGRAMA PRINCIPAL

- Llama a la rutina de inicialización
- Llama repetidas veces a :
 - rutina de tiempo
 - rutina de sucesos

RUTINA DEL TIEMPO

- Determina el siguiente tipo de suceso
- Avanza el reloj de simulación

Rutina de sucesos

- Actualiza el estado del sistema
- Actualiza los contadores
- Genera futuros sucesos y los añade a la lista de sucesos

Librería de rutinas

Genera números aleatorios

¿TERMINÓ LA SIMULACIÓN?

Generador de informes

- Calcula estimaciones de interés
- Escribe el informe

SI

NO

DIAGRAMA DE FLUJO DE LA RUTINA DE LLEGADA DE UN CLIENTE



SUCESO DE LLEGADA

Programar siguiente llegada

Actualizar variables de estado y contadores estadísticos del periodo anterior

SI

¿SERVIDOR OCUPADO?

NO

¿COLA LLENA?

SI

Añadir 1 a clientes perdidos o error si cap. infinita

Poner espera = 0

NO

Añadir 1 a la cola

Añadir 1 al número de clientes servidos

Guardar instante de llegada de este cliente

Poner el servidor ocupado

Programar suceso de salida para este cliente

DIAGRAMA DE FLUJO
DE LA RUTINA DE
SALIDA DE UN
CLIENTE



5. EJEMPLO DE SIMULACION DE UNA COLA CON UN SERVIDOR

Disponemos de los valores aleatorios $A_i =$ Tiempo entre llegadas del cliente $i-1$ e i -ésimo y de $S_i =$ Tiempo de servicio del cliente i -ésimo.

$A_1=0.4, A_2=1.2, A_3=0.5, A_4=1.7, A_5=0.2, A_6=1.6, A_7=0.2, A_8=1.4, A_9=1.9$
 $S_1=2.0, S_2=0.7, S_3=0.2, S_4=1.1, S_5=3.7, S_6=0.6$

OBJETIVO: Tratamos de estimar L_q , W_q y ϕ (o ρ)

$$L_q = \sum_{k=0}^{\infty} k p_k, \text{ siendo } p_k = P(\text{cola haya } k \text{ clientes})$$

• La estimación más razonable para p_k es:

$$\hat{p}_k = \frac{T_k}{T(n)} = \frac{\text{Tiempo en que hay } k \text{ clientes en la cola}}{\text{Tiempo total de simulación}}$$

5. EJEMPLO DE SIMULACION DE UNA COLA CON UN SERVIDOR

• Para estimar W_q , se define D_i = tiempo en cola del cliente i -ésimo. La estimación para W_q será:

$$\hat{W}_q = \frac{1}{n} \sum_1^n D_i, \quad n = n^\circ \text{ total de clientes simulados}$$

• En cada instante, se almacena la suma de los tiempos en cola de todos los clientes (Tiempo total en cola)

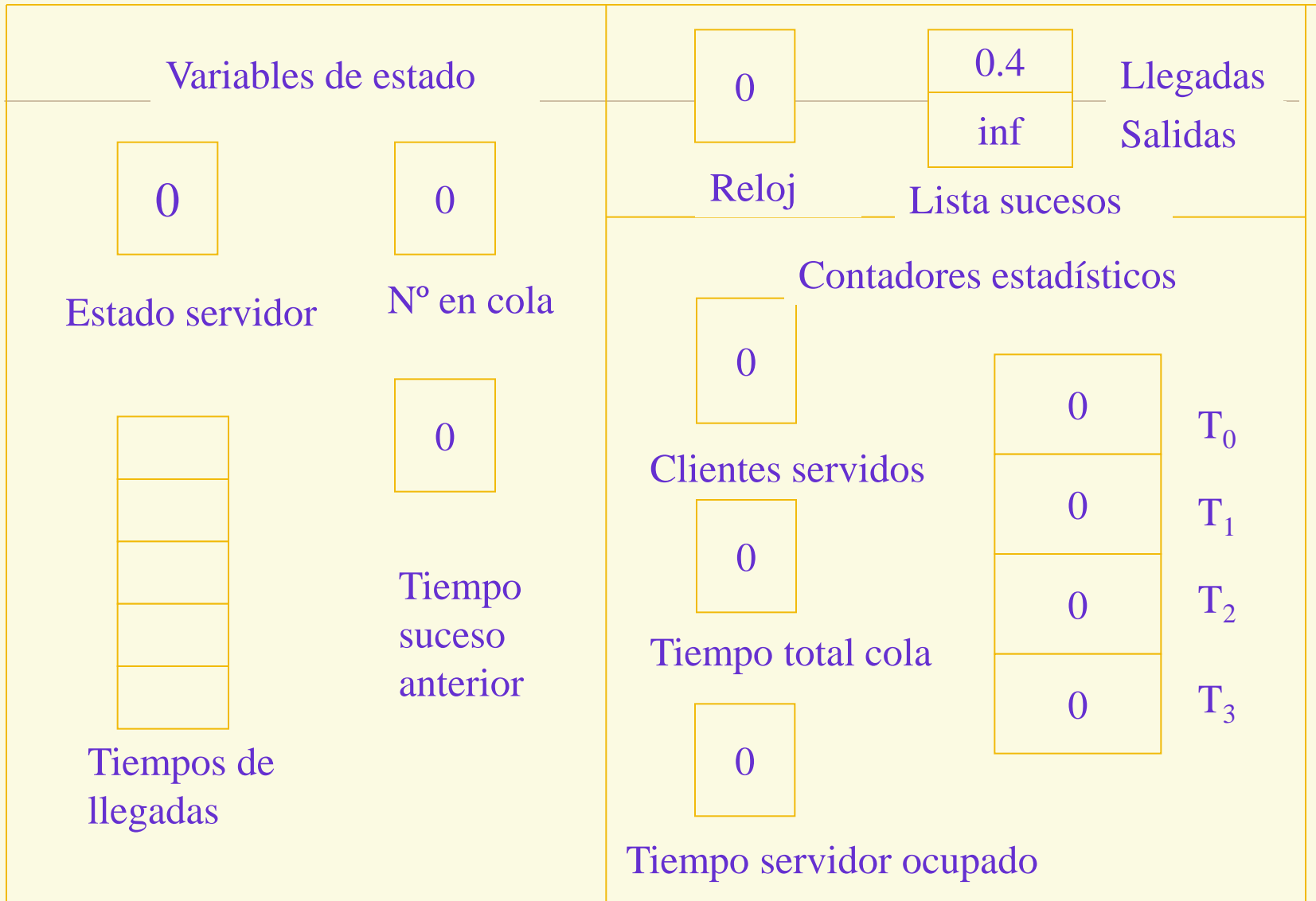
• El criterio de parada: instante en que entra al servidor el sexto cliente

5. EJEMPLO DE SIMULACION DE UNA COLA CON UN SERVIDOR

• Para estimar $\phi = \rho$, se observa el estado del servidor en cada instante (0 si está libre, 1 si está ocupado) y se almacena el tiempo en que el servidor está ocupado. La estimación para ϕ

$$\hat{\phi} = \hat{\rho} = \frac{\text{Tiempo en que el servidor está ocupado}}{T(n)}$$

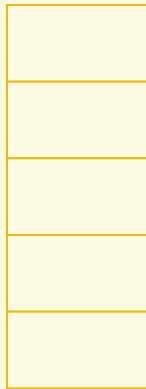
• Estas fórmulas sirven cuando el sistema ha alcanzado el estado estacionario. Queda PENDIENTE decidir cuándo en una simulación se puede decir que se ha alcanzado el estado estacionario.



Variables de estado

1

Estado servidor



0

Nº en cola

0

Tiempo suceso anterior

0.4

Reloj

1.6

2.4

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

1

Clientes servidos

0

Tiempo total cola

0

Tiempo servidor ocupado

0.4

T₀

0

T₁

0

T₂

0

T₃

Variables de estado

1	1
Estado servidor	Nº en cola
1.6	0.4
	Tiempo suceso anterior
Tiempos de llegadas	

1.6

Reloj

2.1

2.4

Llegadas

Salidas

Lista sucesos

Contadores estadísticos

1	1.6	T ₀
Cientes servidos	0	T ₁
0	0	T ₂
Tiempo total cola	0	T ₃
1.2		
Tiempo servidor ocupado		

Variables de estado

1

Estado servidor

1

Nº en cola

2.1

Tiempos de llegadas

2.1

Tiempo suceso anterior

2.4

Reloj

3.8

3.1

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

2

Clientes servidos

0.8

Tiempo total cola

2.0

Tiempo servidor ocupado

1.6

T₀

0.5

T₁

0.3

T₂

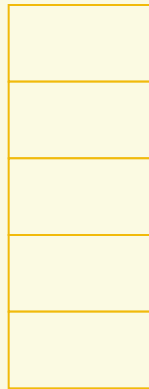
0

T₃

Variables de estado

1

Estado servidor



Tiempos de llegadas

0

Nº en cola

2.4

Tiempo suceso anterior

3.1

Reloj

3.8

3.3

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

3

Clientes servidos

1.8

Tiempo total cola

2.9

Tiempo servidor ocupado

1.6

T₀

1.2

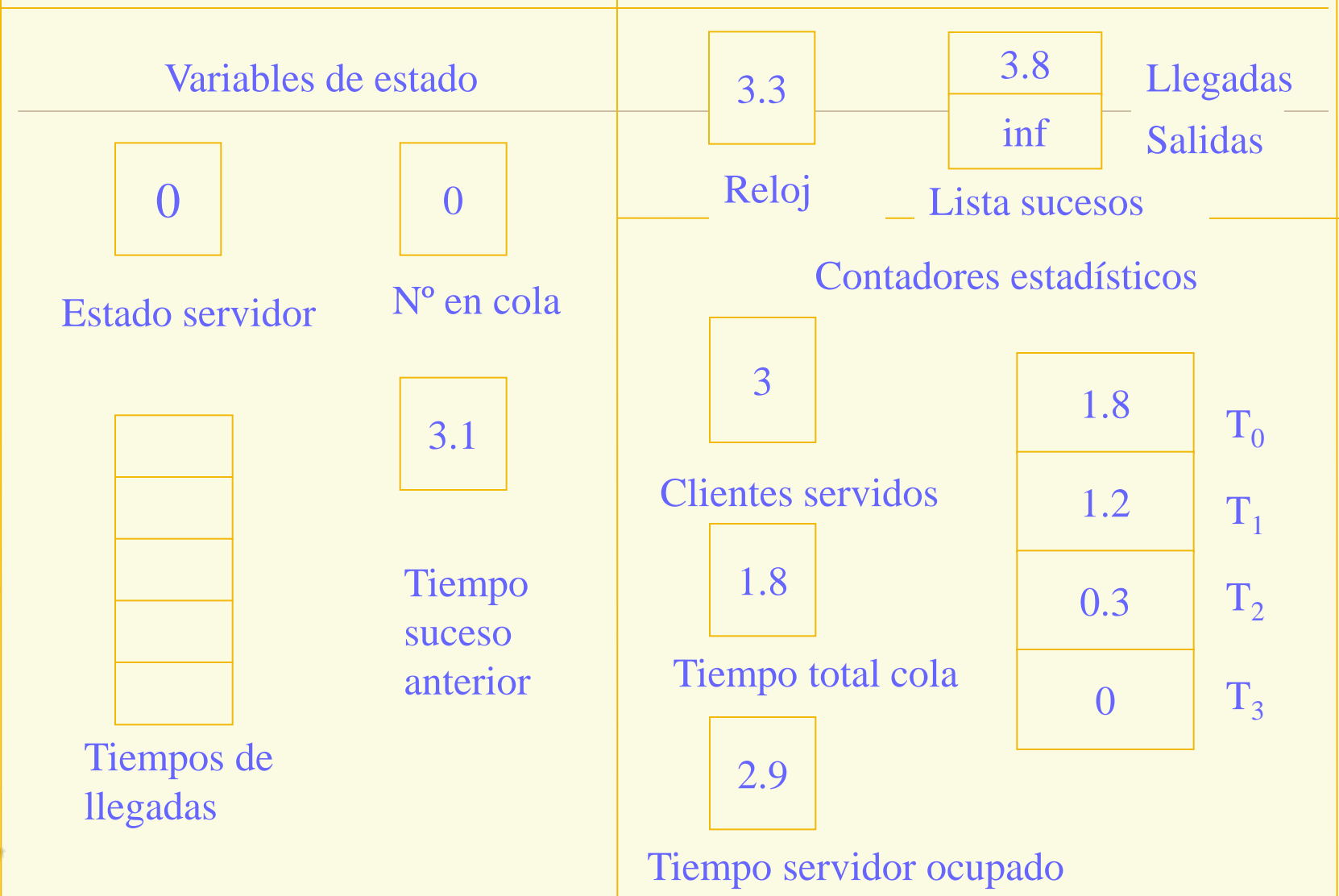
T₁

0.3

T₂

0

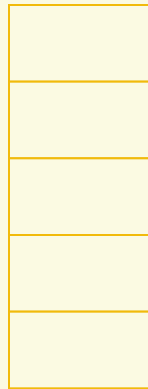
T₃



Variables de estado

1

Estado servidor



Tiempos de llegadas

0

Nº en cola

3.3

Tiempo suceso anterior

3.8

Reloj

4

Clientes servidos

1.8

Tiempo total cola

2.9

Tiempo servidor ocupado

4.0

4.9

Llegadas

Salidas

Lista sucesos

Contadores estadísticos

2.3

T₀

1.2

T₁

0.3

T₂

0

T₃

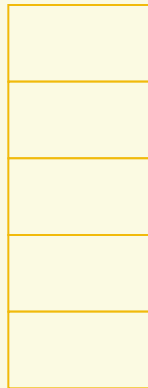
Variables de estado

1

Estado servidor

0

Nº en cola



Tiempos de llegadas

4.0

Tiempo suceso anterior

4.9

Reloj

5.6

8.6

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

5

Clientes servidos

2.5

T₀

2.1

T₁

2.7

Tiempo total cola

0.3

T₂

4.0

Tiempo servidor ocupado

0

T₃

Variables de estado

1	1
Estado servidor	Nº en cola
5.6	4.9
	Tiempo suceso anterior
Tiempos de llegadas	

5.6

Reloj

5.8

8.6

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

5	3.2	T_0
Clientes servidos	2.1	T_1
2.7	0.3	T_2
Tiempo total cola	0	T_3
4.7		
Tiempo servidor ocupado		

Variables de estado

1	2
Estado servidor	Nº en cola
5.6	5.6
5.8	
	Tiempo suceso anterior
Tiempos de llegadas	

5.8

Reloj

7.2

8.6

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

5	3.2	T_0
Clientes servidos	2.3	T_1
2.7	0.3	T_2
Tiempo total cola	0	T_3
6.3		
Tiempo servidor ocupado		

Variables de estado

1

Estado servidor

5.6

5.8

7.2

Tiempos de llegadas

3

Nº en cola

5.8

Tiempo suceso anterior

7.2

Reloj

5

Clientes servidos

2.7

Tiempo total cola

6.3

Tiempo servidor ocupado

9.1

Llegadas

8.6

Salidas

Lista sucesos

Contadores estadísticos

3.2

T₀

2.3

T₁

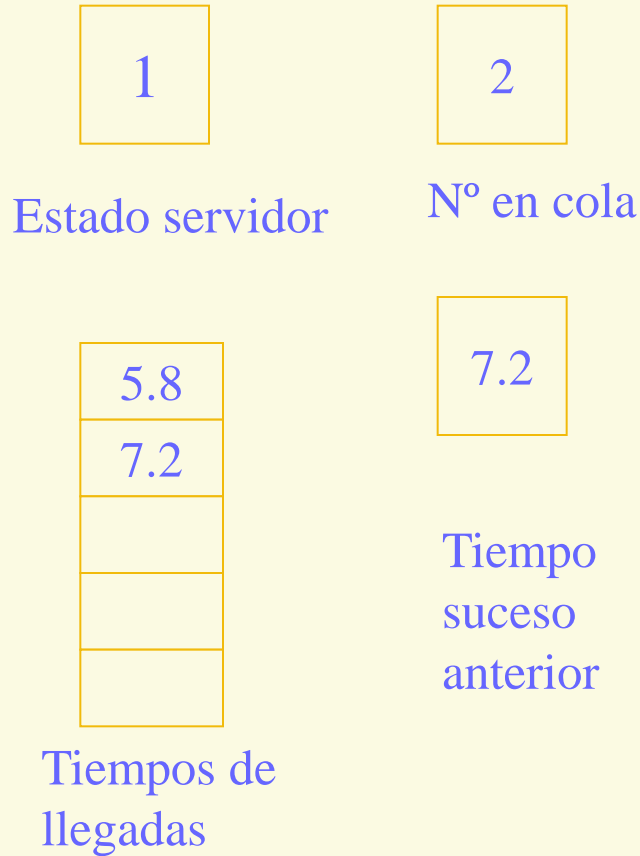
1.7

T₂

0

T₃

Variables de estado



8.6

Reloj

9.1

9.2

Lista sucesos

Llegadas

Salidas

Contadores estadísticos

6

Clientes servidos

5.7

Tiempo total cola

7.7

Tiempo servidor ocupado

3.2

T₀

2.3

T₁

1.7

T₂

1.4

T₃

5. EJEMPLO DE SIMULACION DE UNA COLA CON UN SERVIDOR

Como estimadores se obtienen: $T_0 = 3.2$, $T_1 = 2.3$, $T_2 = 1.7$, $T_3 = 1.4$, $T(n) = 8.6$. Las estimaciones para las probabilidades de equilibrio serán:

$$p_0 = \frac{3.2}{8.6} = 0.3721, \quad p_1 = \frac{2.3}{8.6} = 0.2674,$$

$$p_2 = \frac{1.7}{8.6} = 0.1977, \quad p_3 = \frac{1.4}{8.6} = 0.1628$$

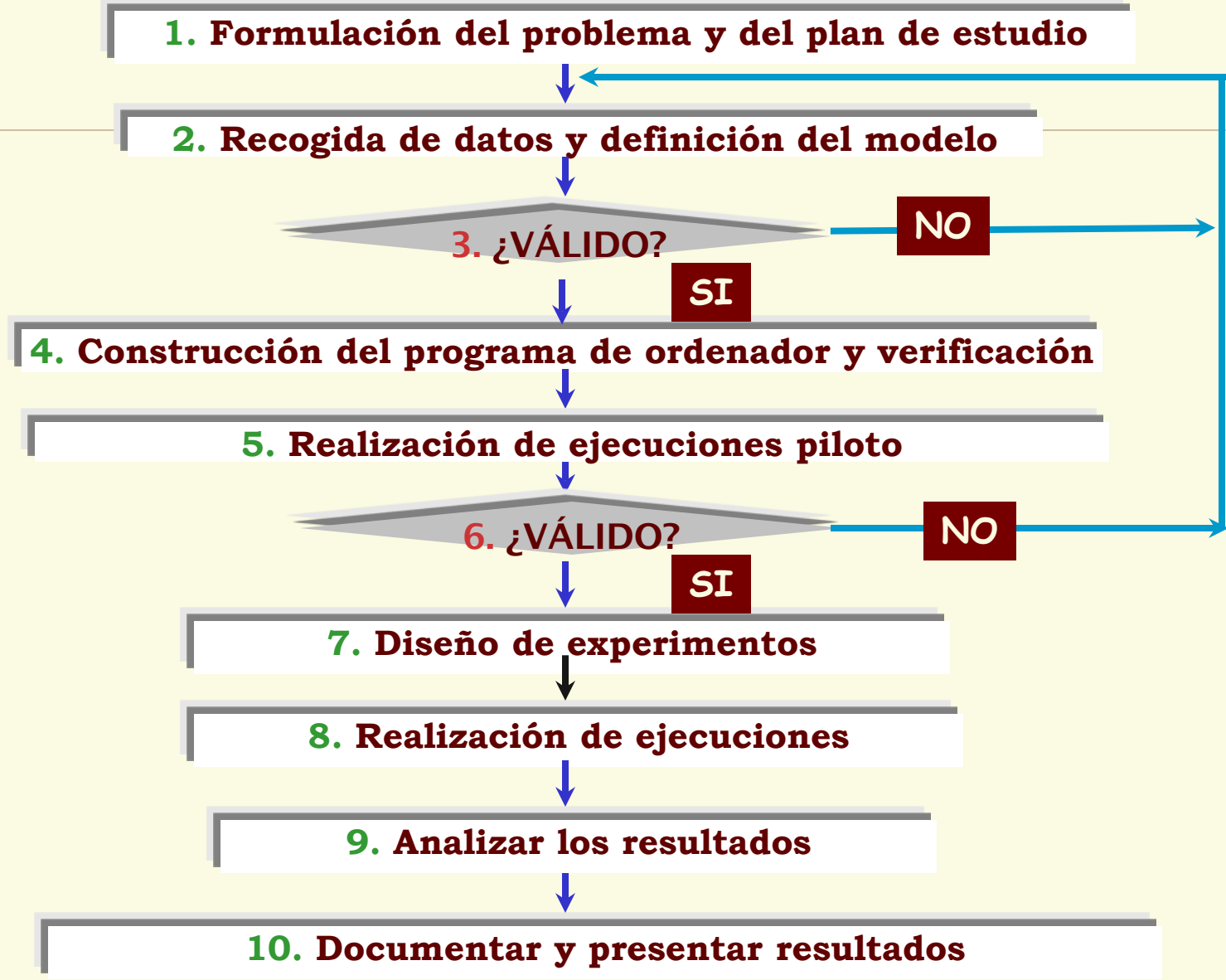
• La estimación de L_q es:

$$L_q = \sum_{k=0}^3 kp_k = 1 \cdot 0.2674 + 2 \cdot 0.1977 + 3 \cdot 0.1628 = 1.1512$$

5. EJEMPLO DE SIMULACION DE UNA COLA CON UN SERVIDOR

- La estimación de $W_q = 5.7/6 = 0.95$ ($n = 6$)
- La estimación de $\phi = \rho = 7.7/8.6 = 0.90$
- Este ejemplo a mano muestra la lógica de programas más complejos.
- Observar que se habrían obtenido resultados diferentes si se repitiese la simulación con otros valores aleatorios.
- **Manera de trabajar:** realizar varias simulaciones y utilizar técnicas estadísticas.
- **Pendiente:** decidir cuántas simulaciones se realizan y de qué longitud.

6. ETAPAS MÁS FRECUENTES DE UN ESTUDIO DE SIMULACIÓN



ETAPA 1: FORMULACIÓN DEL PROBLEMA Y PLAN DE ESTUDIO

- Especificar claramente los objetivos del estudio.
- Indicar las distintas alternativas de diseño a estudiar.
- Decidir qué medidas de rendimiento se van a evaluar para medir la eficiencia de las distintas alternativas.
- Tener en cuenta el coste y tiempo requerido para realizar el estudio.

ETAPA 2: RECOGIDA DE DATOS

➤ Si el sistema existe o existe uno similar al que se quiere simular, los datos recogidos se utilizan para obtener las distribuciones de probabilidad de las v.a. del modelo o validar el modelo posteriormente.

➤ **EJEMPLO:** en un sistema de colas, si recogemos datos entre llegadas consecutivas de clientes y cuánto tiempo en cola esperan los clientes podemos:

➤ Decidir qué distribución continua es más adecuada para modelar la v.a. que mide el tiempo entre llegadas consecutivas.

➤ Validar después el estudio viendo si el tiempo medio en cola obtenido coincide con los datos recogidos.

ETAPA 2: DEFINICIÓN DEL MODELO

- Construir un modelo es más un arte que una ciencia.
- Para construir un modelo conviene:
 - Conocer a fondo el sistema que se quiere modelar.
 - Estar en contacto con las personas que conocen de verdad el sistema que se quiere modelar.
 - Priorizar la lista de cuestiones a estudiar comenzando por un nivel de detalle moderado.

ETAPA 3: VALIDACIÓN

- **VALIDAR** el modelo es tratar de comprobar si el modelo propuesto se adecua al sistema real y al objetivo perseguido.
- La validación debe de realizarse a lo largo de todo el estudio.
- En este momento, se puede:
 - Contrastar mediante técnicas estadísticas las distribuciones de probabilidad propuestas.
 - Ponerse en contacto con las personas que conocen el sistema para que opinen sobre el modelo propuesto.

ETAPA 4: CONSTRUCCIÓN DEL PROGRAMA DE ORDENADOR Y VERIFICACIÓN DEL MISMO

- Hay que decidir en qué lenguaje se programa: de tipo general, específico o si se va a utilizar un simulador.
- La VERIFICACIÓN del programa consiste en comprobar que el programa de simulación implementa correctamente el modelo conceptual de simulación.
- **EJEMPLO:** en colas, al implementar un sistema M/M/1/5, hay que ver si los resultados que se obtienen mediante el programa coinciden con los resultados analíticos que hemos estudiado para este sistema.

ETAPA 4: CONSTRUCCIÓN DEL PROGRAMA DE ORDENADOR Y VERIFICACIÓN DEL MISMO

- Para la verificación se utilizan las técnicas habituales de depuración y verificación de programas.
- **Técnicas a utilizar:**
 - Escribir y depurar los programas en módulos y subprogramas
 - Correr la simulación con distintos conjuntos de parámetros de entrada, y comprobar que la salida es razonable
 - Depurar con la técnica de traza.
 - Correr el modelo con hipótesis simplificadas para las que el modelo tenga solución analítica.

ETAPAS 5 Y 6: REALIZACIÓN DE EJECUCIONES PILOTO Y VALIDACIÓN

- Se realizan varias ejecuciones de prueba y los resultados nos servirán para validar el modelo.
 - Si existe un sistema similar al que se quiere estudiar: comparar los datos de salida del sistema, y los del modelo de simulación. Si son similares, el modelo es válido. Después se modifica el modelo para representar al sistema.
 - Si no existe un sistema similar: se intenta simplificar el modelo de forma que tenga solución analítica, y después se modifica el modelo para representar al sistema real.

ETAPA 7: DISEÑO DE EXPERIMENTOS

➤ Las técnicas estadísticas de Diseño de Experimentos se utilizan cuando hay muchas alternativas de diseño y no se pueden simular todas, porque son demasiadas configuraciones de sistemas. Se decide entonces qué alternativas se simulan y cuales no para obtener la óptima.

ETAPA 8: REALIZACIÓN DE EJECUCIONES

- La salida de una simulación es aleatoria por lo que hay que realizar varias ejecuciones para estimar las características de interés mediante técnicas estadísticas.
- Hay que decidir cuántas ejecuciones se realizan y la longitud de las mismas.

ETAPA 9: ANÁLISIS DE RESULTADOS

- Para analizar los resultados y obtener conclusiones fiables se utilizan técnicas estadísticas.
- Para realizar la estimación de las medidas de interés (parámetro), usaremos la **MEDIA** de los resultados obtenidos en cada simulación.
- Para medir el **ERROR** en la estimación de cada característica, construiremos un **INTERVALO DE CONFIANZA** para la misma.

ETAPA 10: DOCUMENTACIÓN Y PRESENTACIÓN DE RESULTADOS

- Es importante documentar tanto el programa como las hipótesis hechas y los resultados obtenidos.
- Si posteriormente se quieren introducir cambios (otras hipótesis, diferente nivel de detalle,...) se podrá aprovechar el estudio ya realizado.
- Es fundamental para presentar a las personas que han encargado el estudio los resultados del mismo.

ETAPA 4: ELECCION DEL LENGUAJE DE PROGRAMACION. SIMULADORES. ANIMACION

LENGUAJES GENERALES: C es el más utilizado en simulación

LENGUAJES ESPECÍFICOS: Simescript II.5, GPSS, Slam, Simula, Simulink, ModSim III, Extend,...

VENTAJAS DE LENGUAJES ESPECIFICOS FRENTE A GENERALES:

- Programas más cortos:
 - Están preparados para llevar la contabilidad de los distintos parámetros
 - Tienen módulos para generar números aleatorios de las distintas distribuciones, reloj, etc

ETAPA 4: ELECCION DEL LENGUAJE DE PROGRAMACION. SIMULADORES. ANIMACION

- Al tener menos líneas de código:
 - Es más fácil detectar posibles errores
 - Se reduce el tiempo de programación
- Existen una serie de errores típicos que suelen ser identificados y chequeados de forma automática

ETAPA 4: ELECCION DEL LENGUAJE DE PROGRAMACION. SIMULADORES. ANIMACION

INCONVENIENTES DE LENGUAJES ESPECIFICOS FRENTE A GENERALES:

- Son menos flexibles, y su sintaxis tiende a ser menos natural
- Las ejecuciones son más lentas siempre y cuando el programa en lenguaje general esté diseñado de forma eficiente. Esto se debe a que suelen incluir gran variedad de sistemas y no van al caso concreto que se trata.
- Son menos conocidos por analistas y programadores,.

SIMULADORES

Son paquetes que permiten simular algunos tipos de sistemas con poca o ninguna necesidad de programar. Los sistemas se seleccionan a base de menús y de gráficos.

Ventaja: Ahorran tiempo de programación

Inconvenientes:

- Están limitados a modelizar sólo las configuraciones de sistemas que permite el simulador.
- Las ejecuciones son más lentas

Los más conocidos son:

- Simfactory II.5
- Simulink (Matlab)
- Witness
- Comnet III (específico redes de comunicaciones)
- QSIM (incluido en Wingsb)
- Xcells+

ANIMACION

Los elementos clave de un sistema se representan mediante iconos que cambian de forma, color o posición cuando hay un cambio de estado en la simulación.

Modos de operar:

- **En directo:** Se desarrolla la animación mientras se está corriendo la simulación.
- **En diferido:** La animación se visualiza una vez completada la simulación.

VENTAJAS E INCONVENIENTES DE LA ANIMACION

Ventajas:

- Sirven para comunicar la esencia del modelo de simulación a los directivos.
- Puede ayudar a entender el comportamiento dinámico del sistema.

Inconvenientes:

- Aumenta el tiempo para desarrollar el programa de simulación.
- La ejecución es muy lenta.
- No puede sustituir a un cuidadoso análisis estadístico de los resultados.

7. VENTAJAS DE LA SIMULACION

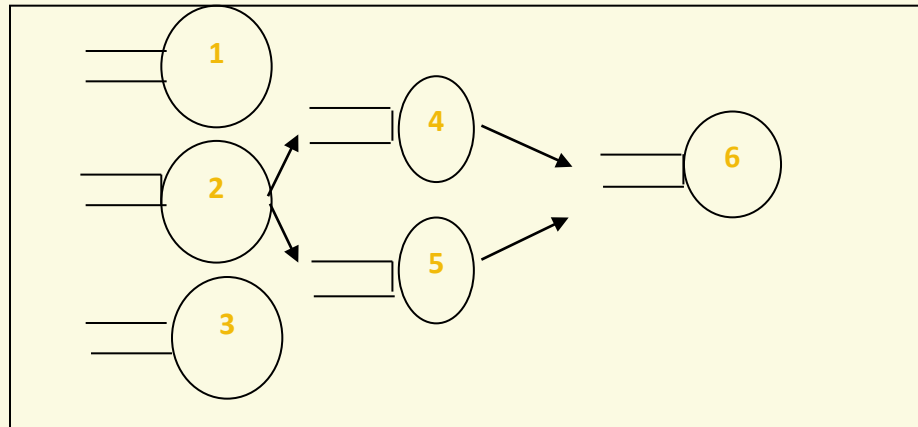
- Permite estudiar sistemas reales que no se pueden evaluar analíticamente
- Hace posible estimar el comportamiento de un sistema existente si se modifican algunas de las condiciones de funcionamiento actuales
- Se pueden comparar distintas alternativas de diseño (o de formas de operar de un sistema)
- Permite estudiar en poco tiempo la evolución de un sistema en un periodo largo de tiempo y al revés
- Se puede utilizar para validar un modelo analítico

7. DESVENTAJAS DE LA SIMULACION

- No produce resultados exactos, sino estimaciones. Esto hace necesario el uso de técnicas estadísticas
- Desarrollar un modelo de simulación suele ser caro y lleva tiempo
- Es difícil demostrar la validez del modelo.
- Es difícil encontrar el óptimo: sólo se puede encontrar el mejor entre varias alternativas

5.-TRABAJO DE SIMULACION 2015

📄 **OBJETIVO:** simular el comportamiento de esta red de colas



DISTRIBUCIONES DE TIEMPO ENTRE LLEGADAS Y TIEMPO DE SERVICIO

- ☞ Hay que generar números aleatorios de las distintas distribuciones, exponenciales, Erlang y variables discretas.
- ☞ Cada vez que se necesita generar una llegada al sistema, un tiempo de servicio, o indicar a donde se dirige un mensaje que sale de un nodo, hay que llamar a las subrutinas de generación de números aleatorios.

Distribución exponencial



Si se desea generar un número aleatorio x de una exponencial con parámetro β ,

1. Generar un número aleatorio de $U(0,1)$ como $u := \text{random}$; (en Pascal)

$u = \text{rand()} / (\text{double})\text{RAND_MAX}$ (en C)
(comprobar que es distinto de cero)

- Aplicar la fórmula :

$$x := \frac{-1}{\beta} \ln(u)$$

Distribución Erlang



Si se desean generar un número aleatorio x de una Erlang (n, β) ,

1. Generar n números aleatorios de $U(0,1)$ como $u_i := \text{random}, i = 1, \dots, n$; (en Pascal)
2. Aplicar la fórmula :

$$x := \frac{-1}{\beta} \ln \prod_{i=1}^n u_i$$

Distribución Erlang



Si se desean números aleatorios de una Erlang de parámetros α , β

- 1. Generar α números aleatorios de $U(0,1)$ como antes;
- 2. Aplicar la fórmula :

$$x := \frac{-1}{\beta} \log \prod_{i=1}^{\alpha} u_i$$

Distribución discreta



Para decidir si un cliente que sale del nodo 1,2,o 3 va al nodo 4 o 5:

- 1.- Generar un número aleatorio de $U(0,1)$ como antes;
- 2.- Si $u < 0.5$, ir al nodo 4,
si $u \geq 0.5$, ir al nodo 5.

Nota: En el programa principal ejecutar una sola vez:

```
srand(time(NULL));
```

para que el generador de aleatorios utilice distinta semilla cada vez que se ejecuta el programa.

TRABAJO DE SIMULACION CURSO 2015

VARIABLES DE ESTADO: Son aquellas que describen el sistema en un instante dado

Estadoservidor 1..6: libre, ocupado;

Numcola 1..6: integer;

Tiemposllegadas, Tiemposllegadas6 :array .. of real;

Reloj, Relojtrans: real;

Suceso anterior: real;

CONTADORES ESTADÍSTICOS

Variables que se usan para guardar la información del sistema que luego permitirá hacer las estimaciones que se piden.

Tiempo0-2clientesnodo4:real

Tiempoclientesnodo6:array [0..10] of real

Tiempototalnodo6: real

**Clientesperdidosnodo6, Clientesservidosnodo6:
integer**

Tiempototalsistema:real

LISTA DE SUCESOS

SUCESOS: ocurrencia que puede modificar el estado del sistema

Llegadas a los nodos 1, 2, 3

Salidas de nodos 1, 2, 3 = llegadas a nodos 4, 5

Salidas de nodos 4, 5 = llegadas a nodo 6

Salidas de nodo 6 al exterior

Lista de sucesos: guarda los instantes en los que se producen los próximos sucesos: array [1..9]

RUTINA DE TIEMPO

Suceso anterior = Reloj;

Determinar el próximo suceso (mínimo de la lista de sucesos);

Actualizar reloj (instante en que se produce el próximo suceso);

Actualizar los CONTADORES ESTADÍSTICOS:

Tiempo0-2clientesnodo4

Tiempoclientesnodo6:array [0..10] of real

Es decir, se suma (reloj - relojant) al primer contador si hay menos de 3 clientes, y al segundo contador dependiendo de cuantos clientes haya en el nodo 6.

Devuelve el control al programa principal que se encarga de llamar a la correspondiente rutina de sucesos.

DIAGRAMA DE FLUJO DE LA RUTINA DE LLEGADA DE UN CLIENTE A AGUNO DE LOS NODOS 1, 2, 3.

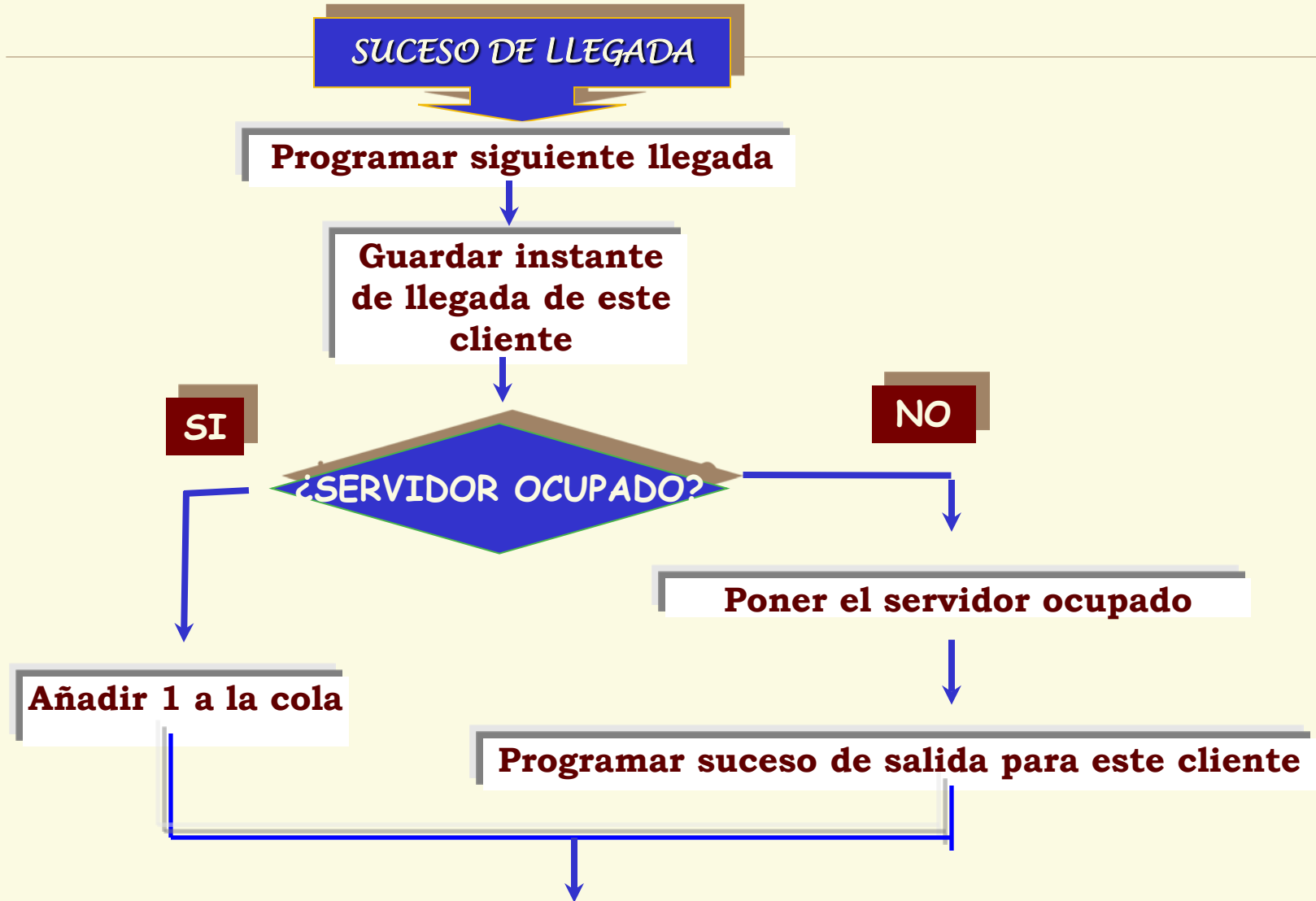


DIAGRAMA DE FLUJO DE LA RUTINA DE SALIDA DE LOS NODOS 1, 2 o 3

SUCESO DE SALIDA

Decidir si va al nodo 4 o al 5

Aplicar lo indicado en el diagrama de llegada desde ¿servidor ocupado?

SI

¿COLA VACÍA?

NO

Poner libre el servidor

Restar 1 al número en la cola

Dejar de considerar el suceso de salida (poner infinito en la lista de sucesos)

Programar el suceso de salida para el cliente que entra al servidor

DIAGRAMA DE FLUJO DE LA RUTINA DE SALIDA DE LOS NODOS 4 o 5

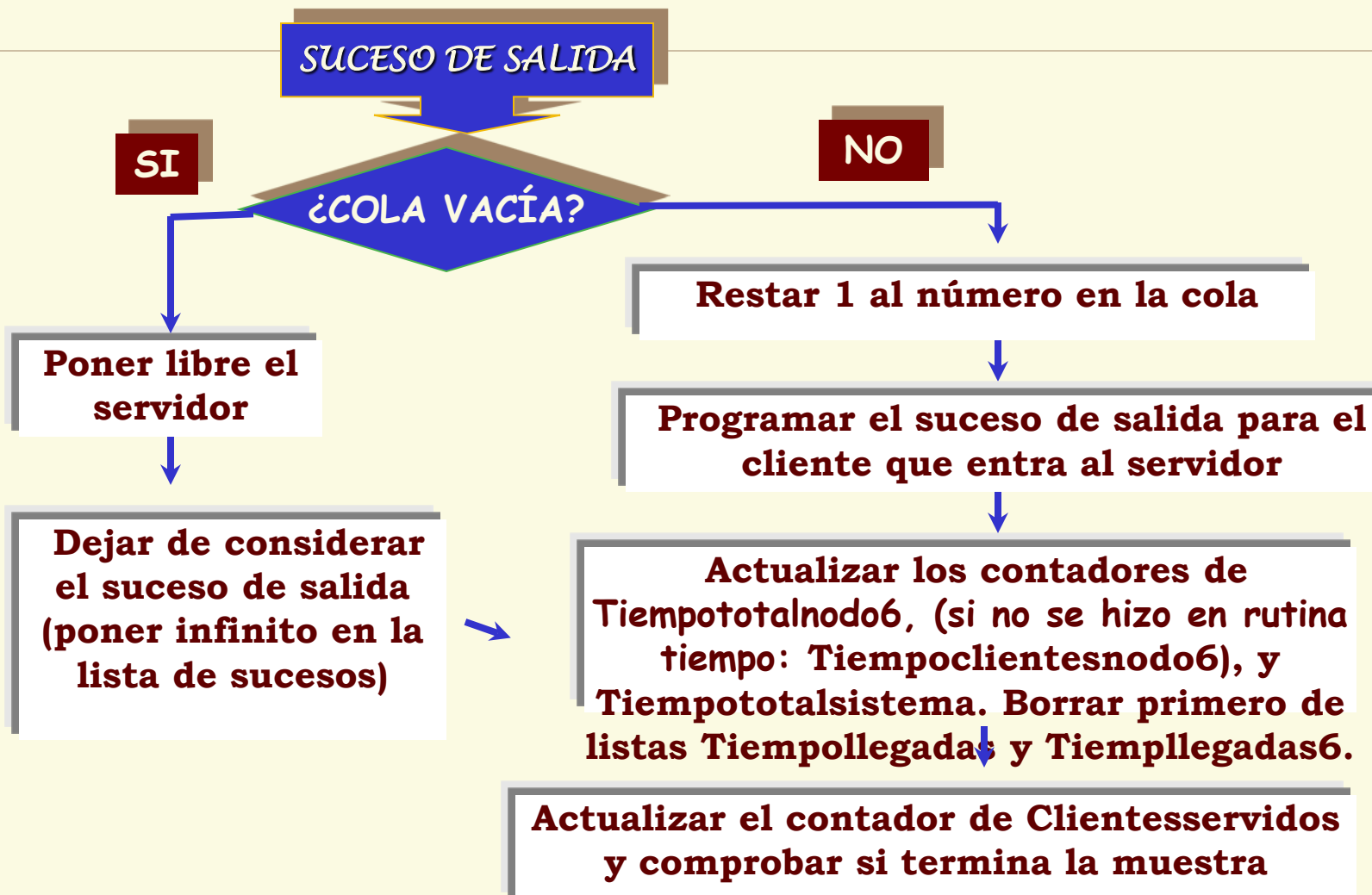
SUCESO DE SALIDA

Si es el nodo 4 y había menos de 3 clientes, actualizar el contador correspondiente (si no se ha hecho en la rutina de sucesos)

Si la cola del nodo 6 está llena, actualizar contadores Tiempototalsistema (borrar uno de lista llegadas) y clientesperdidos. Si no llena, aplicar diagrama de llegada desde Guardar instante de llegada.



DIAGRAMA DE FLUJO DE LA RUTINA DE SALIDA DEL NODO 6



PARA CADA MUESTRA

- Al terminar el periodo transitorio: $\text{Relojtran} = \text{Reloj}$ y ponemos a cero los contadores estadísticos, pero no las variables de estado.
- En la ejecución del programa no se darán los resultados de cada muestra.
- Debe de ser posible detener la simulación en todo momento.
- Al terminar cada muestra, realizamos la estimación de las características de interés que guardaremos para hacer las estimaciones finales a partir de los resultados obtenidos para las 25 muestras.

ESTIMACIONES EN CADA SIMULACIÓN

- Al final del programa, para calcular Probabilidades dividiremos los contadores entre (Reloj-Relojtran).

$$p_k = \frac{\text{Tiempo sistema ha habido } k \text{ clientes}}{\text{Reloj-Relojtran}}$$

- El número medio de clientes en el sexto nodo:

$$L = \sum_{k=0}^{10} k \cdot p_k$$

- El tiempo medio que un cliente pasa en el sexto nodo:

$$W = \frac{\text{Tiempototalnodo6}}{\text{n}^\circ \text{ total clientes estado estacionario (10.000)}}$$

ESTIMACIONES EN CADA SIMULACIÓN

- Para estimar el número de clientes perdidos en el nodo 6:

$\text{Clientes perdidos} / (\text{Clientes perdidos} + \text{Clientes servidos}_6)$.

- Para estimar W total en el sistema:

$$W = \frac{\text{Tiempo total sistema}}{\text{n}^\circ \text{ total clientes } \textit{servidos} \text{ estado estacionario} + \text{perdidos}}$$

ESTIMACIÓN PARA L

• Estimación de L: Se realiza la media entre los valores obtenidos en todas las simulaciones, L_i .

$$\hat{L} = \frac{1}{n} \sum_{i=1}^n L_i. \text{ Antes hemos definido } n = 25.$$

• Intervalo para L es:

$$\left(\hat{L} \pm h \frac{S}{\sqrt{n}} \right)$$

• En el trabajo, $h = 2.06$ (95%), $h = 1.71$ (90%) y S se obtiene:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (L_i - \hat{L})^2 = \frac{1}{n-1} \sum_{i=1}^n L_i^2 - \frac{n}{n-1} \hat{L}^2$$

ESTIMACIONES PARA L , p_k y W

- El error relativo para la estimación de L es:

$$\text{Error relativo: } h \frac{S}{\sqrt{n}} / \hat{L}$$

- Las estimaciones para las probabilidades de que haya k clientes en el sistema, p_k , y W se obtienen de manera análoga, calculando la media aritmética de los resultados obtenidos en las 25 ejecuciones.
- El intervalo de confianza y el error relativo para W se obtienen de manera análoga al obtenido para L .

VERIFICACIÓN DEL PROGRAMA

- Con tiempo de servicio exponencial, tenemos una red de Jackson (con resultados analíticos).
- El nodo 4 es se comporta como un nodo M/M/1 con tasa de llegadas $\lambda_4 = 1.2$ y tasa de servicio $\mu = 2$. Los resultados exactos para este sistema se pueden obtener con el procedimiento QA de WinQSB.
- El nodo 6 es se comporta como un nodo M/M/1/10 con tasa de llegadas $\lambda_6 = 2.4$ y tasa de servicio $\mu = 5$. Los resultados exactos para este sistema se pueden obtener con el procedimiento QA de WinQSB.
- Es condición necesaria y suficiente para aprobar que los resultados del programa con tiempos entre llegadas y tiempos de servicio exponenciales **SEAN COHERENTES** con los resultados analíticos obtenidos para los nodos 4 y 6 mediante QA.