

# Simulación Computacional

## Tema 2: Simulación de sistemas

Irene Tischer



Escuela de Ingeniería y Computación  
Universidad del Valle

# Contenido

- 1 Introducción
- 2 Conceptos de la simulación por eventos discretos
- 3 El estudio completo de simulación

## '?Qué es simulación?

Definición: "Una simulación es la imitación del comportamiento de un proceso o sistema del mundo real". (Banks et al., 2001)

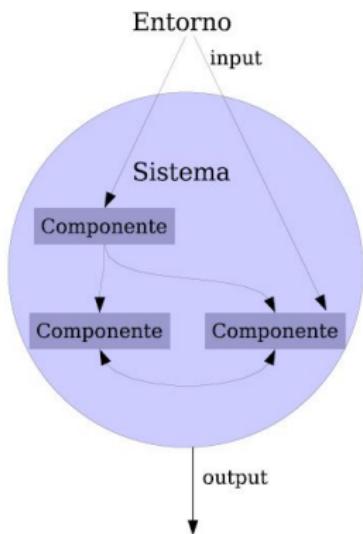
- Por el modelo de simulación se genera una historia artificial del sistema y se infiere a las características operativas del sistema real
- La simulación es una herramienta para la toma de decisiones; se aplica si se presenta un problema en el sistema real.

# Definición de "Sistema"

**Sistema:** Colección de componentes (entidades) que interactúan entre sí para lograr un objetivo común. Un sistema tiene un input (desde el entorno) y un output. El entorno puede afectar el sistema, pero el sistema no afecta el entorno (esta exigencia ayuda en la definición del sistema).

**Funcionamiento del sistema:** Dado por un conjunto de supuestos expresados como relaciones matemáticas, lógicas o simbólicas entre las entidades.

**Dinámica (comportamiento) del sistema:** Se deriva de la estructura y del comportamiento.



# Modelo

Un modelo es la representación de un sistema, hecho para facilitar su estudio

- mg

Es una simplificación del sistema real

Debe ser suficientemente detallado para permitir conclusiones del modelo al sistema (esto obliga a validar el modelo: se debe determinar que tan cerca se encuentra a la realidad)

OJO: de acuerdo del punto de vista y del problema a resolver existen una variedad de posibles modelos para cada sistema

## Tipos de modelo

**Modelos matemáticos:** usan notación simbólica, ecuaciones, funciones

Modelos físicos: maquetas, mapas

Modelos determinísticos: cada entrada produce una única salida

**Modelos estocásticos:** una entrada puede conducir a diferentes salidas

Modelos continuos: las variables principales son continuas

**Modelos discretos:** las variables principales toman valores discretos

Modelos estáticos: son independiente del tiempo

**Modelos dinámicos:** tienen en cuenta el desarrollo del sistema/a lo largo del tiempo

MEDD

Un modelo de simulación de sistemas de eventos discretos se clasifica como modelo matemático, estocástico discreto y dinámico.

# Proceso de simulación

- Analizar el sistema real y el problema que se presenta
- Traducirlo en un modelo de simulación computacional
- Observar el sistema simulado (los datos que arroja) en diferentes escenarios
- Concluir para estimar el desempeño del sistema real

Estructura

Algoritmo

# Simulación en aplicaciones

- Mejoramiento del desempeño tecnológico (ejemplos: una fábrica de montaje de automóviles; red de computadores con sus usuarios, el tráfico en una cruce de calles con semáforos; urgencias de un hospital)
- Ingeniería de seguridad (reducir el riesgo en sistemas, diseñar sistemas con seguridad alta)
- Vídeo-juegos (se requieren escenarios realistas con suficiente variación, en dependencia de acciones del usuario)
- Educación (e-learning, aprendizaje en ambientes virtuales)
- Ciencias (simulaciones de entender mejor un fenómeno por ejemplo en: economía, ecología, análisis climático, biología molecular)

## Ventajas de simulación

- Simulación es más seguro y barato que la experimentación con el sistema real.
- A veces la simulación es única posibilidad de experimentar (sistemas en proceso de desarrollo, sistemas donde la experimentación disminuye la seguridad).
- Por ser estocásticos, modelos de simulación son más realistas que los modelos determinísticos.
- Permiten tener en cuenta diferentes aspectos a mejorar.
- Es más fácil desarrollarlos que desarrollar modelos de programación matemática.
- Es más fácil entenderlos que entender modelos de programación matemática.
- El costo de desarrollo es bajo, ya que los componentes de los modelos de simulación son reutilizables.

## Desventajas de simulación

- Los modelos de simulación no generan soluciones óptimas; sólo facilitan sólo la generación de soluciones buenas, que pueden tener en cuenta muchos aspectos.
- Las variaciones propuestas no deben alejarse demasiado de la situación actual, ya que mucha debido a demasiado variación se puede perder la validez del modelo.

# Enfoques de simulación

Hay básicamente 2 enfoques para la construcción de modelos de simulación:

- **la simulación orientada a procesos**

- ventaja: más cercano a las definiciones formales del sistema, código más compacto
- desventaja: los procesos son difíciles de sincronizar

- **la simulación orientada a eventos**

- ventaja: generalmente es más rápido, no hay problema de sincronización
- hay que tener en cuenta el orden in que se realizan los eventos

Aquí se considera sólo la simulación conducida por eventos

# Ejemplo

## El problema

En una sala de cine se controlan las boletas en la puerta de entrada. La gente llega (el tiempo entre llegadas es entre 0 y 120 segundos) y hace cola para poder entrar. El control de la boleta requiere  $70 \pm 30$  segundos. Son 100 personas quienes quieren ver la película. ¿Cuánto tiempo necesitan las personas a entrar?

Simule el sistema hasta que todas las personas entraron.

Analizar el tiempo que se demoraron las personas a entrar.

?Se forma una cola?

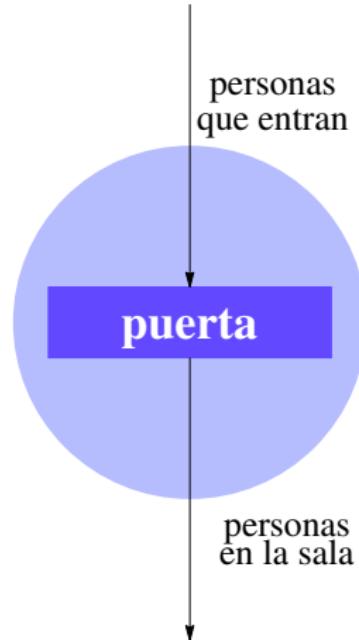
?Que es el tiempo promedio de espera en la cola?

Repite la simulación para un tiempo de control de entrada de  $50 \pm 30$  segundos.

?Conclusiones?

# Ejemplo

## El sistema



El sistema es muy sencillo:

- única componente: la puerta
- entrada: personas que llegan
- salida: personas en el estadio ~~Cine~~

~~Cine~~

Funcionamiento:

- la persona llega y hace cola
- se revisa su boleta
- entra a la sala de cine

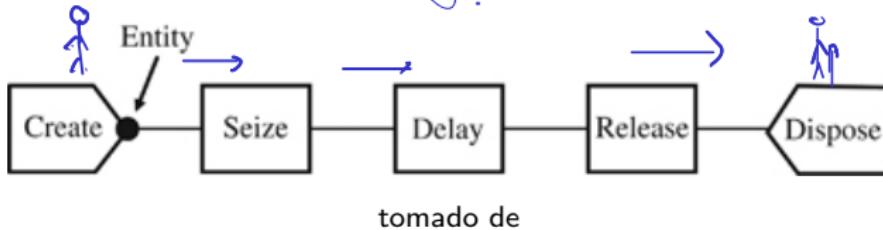
Comportamiento (?):

- cola que se forma
- tiempo de espera

# Ejemplo

## Simulación orientada a procesos

Se tiene un tipo de proceso, que corresponde a la manera como se mueve una persona por el sistema. Para cada persona se crea un nuevo proceso, el proceso asume control cuando la persona comienza la atención en la puerta, se demora mientras que la persona está atendida, luego se destruye.



tomado de

[http://www.palgrave-journals.com/jos/journal/v6/n1/fig\\_tab/jos201027a.html](http://www.palgrave-journals.com/jos/journal/v6/n1/fig_tab/jos201027a.html)

La sincronización de los procesos está hecha por el desarrollador de modelo, afuera de los procesos.

# Ejemplo

## Simulación orientada a eventos (1)

Se modelan los cambios del sistema. Las variables que describen el sistema en un momento dado son las **variables de estado**:

- Estado del controlador de los boletas (**servidor**): puede estar ocupado o desocupado
- Estado de la **cola**: número de personas esperando.

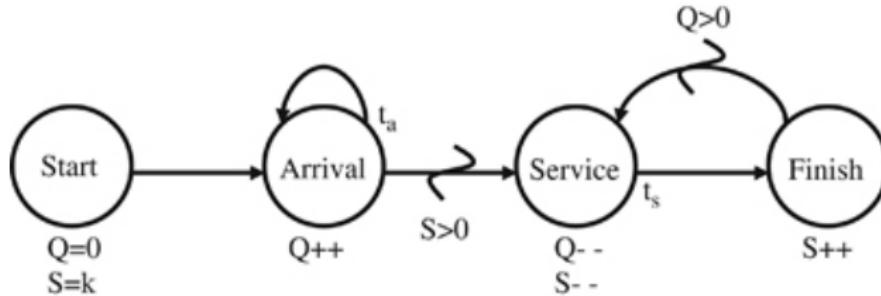
Los cambios se pueden detectar analizando cómo se mueve una persona por el sistema (**actividad**). Actividades son los elementos de la simulación, que le dan la dinámica (sin personas-actividades, nada se mueve). Las variables que son responsables para el cambio del sistema son las **variables de entrada**, el tiempo entre llegadas y el tiempo de atención.

# Ejemplo

## Simulación orientada a eventos (2)

Hay 2 eventos que cambian el estado del sistema:

- Evento de Llegada: Llega una nueva persona; de acuerdo con el estado del servidor entra a la cola o será atendido.
- Evento de Salida: El servidor termine la atención a una persona; de acuerdo con el estado de la cola atiende a la próxima persona en la cola o se pone desocupado.



tomado de

## Introducción

Conceptos de la simulación por eventos discretos  
El estudio completo de simulación

# Ejemplo

## La simulación

Random1	Random2	persona	tiempo entre llegadas (seg)	hora de llegada (seg)	cola al llegar	comienza servicio	tiempo de atención (seg)	hora de salida (seg)	tiempo espera	cola max.
0.7126		1	0	0	0	0	83	83	0	0
0.3231	0.4591	2	39	39	1	83	68	151	44	1
0.0795	0.4789	3	10	49	2	151	69	220	102	2
0.1375	0.9765	4	17	66	3	220	99	319	154	3
0.7422	0.0799	5	89	155	2	319	45	364	164	3
0.2796	0.2249	6	34	189	3	364	53	417	175	3
0.9236	0.2044	7	111	300	3	417	52	469	117	3
0.7183	0.9862	8	86	386	2	469	99	568	83	3
0.2172	0.0528	9	26	412	3	568	43	611	156	3
0.3274	0.8701	10	39	451	3	611	92	703	160	3
0.424	0.5013	11	51	502	3	703	70	773	201	3
0.0557	0.2276	12	7	509	4	773	54	827	264	4

# Ejemplo

## Escenarios

$$\cancel{70 \pm 30} \rightarrow 50 \pm 30$$

Todo la simulación se hace para buscar soluciones a un problema, en el caso de este ejemplo la formación de colas largas en la entrada a la sala de cine.

Posibles alternativas de soluciones al problema son:

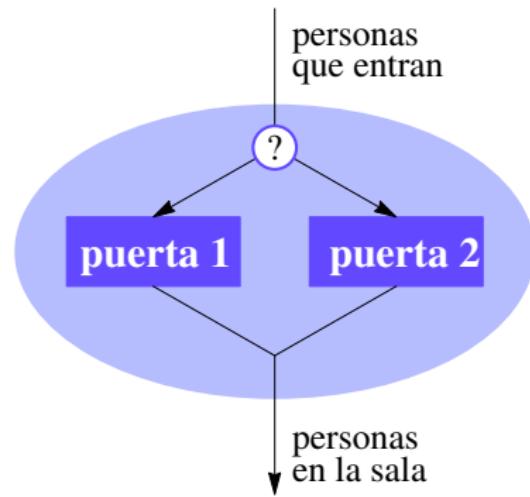
- ① abrir una segunda puerta
- ② acelerar la revisión de las boletas (un segundo controlador, más entrenamiento, optimización del proceso de revisión,...)

Estas soluciones se traducen a **escenarios** de simulación.

Se simulan los escenarios y comparan los resultados.

# Ejemplo

## Escenario 1

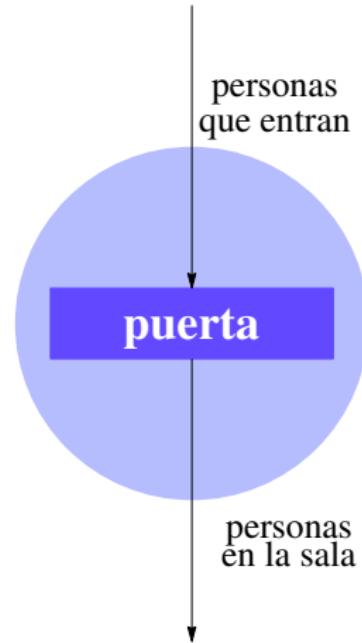


Los cambios al modelo de simulación son pocos:

- se introduce una segunda puerta
- Las personas que entran, deben decidirse para una de las dos puertas

# Ejemplo

## Escenario 2



El único cambio que requiere el modelo de simulación es modificar el tiempo de atención.

# Ejemplo

## Análisis de los resultados

Se quiere saber en qué medida se mejora el sistema cuando se introducen las posibles alternativas.

Por eso se debe evaluar los resultados de cada escenario y comparar los resultados.

Las variables que son importantes para ver como se comporta el sistema, se llaman **variables de desempeño**.

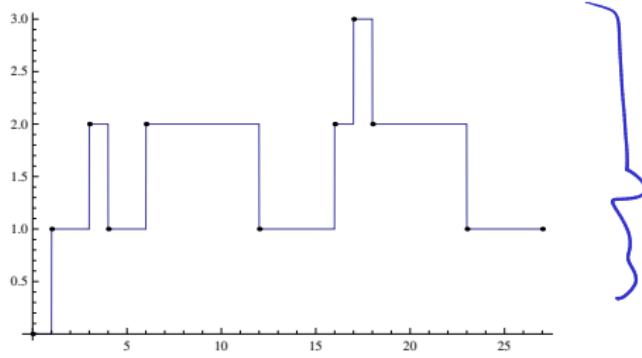
En el caso del cine se podría analizar:

- comportamiento de la cola;
- tamaño promedio de la cola;
- tiempo promedio de espera en cola.

## Variables de estado discretas

- Un modelo de simulación se construye a través de variables aleatorias.
- En la simulación de eventos discretos, las variables de estado son discretas.

Ejemplo: el comportamiento de una cola



# Cambios discretos

Entre dos cambios de las variables de estado, el sistema no cambia.

Los eventos son responsables para un cambio del estado del sistema

⇒ eventos discretos

Por eso, la forma más económica de simular un sistema de cambios discretos es, sólo acceder y ajustar el sistema cuando ocurren los cambios

⇒ simulación de eventos discretos

# Actividades

- Las actividades son las entidades que dan la dinámica al sistema.
- Son entidades que se mueven por el sistema y producen eventualmente cambios de las variables de estado que se pueden asociar con eventos.
- Las actividades son relacionadas con las variables de entrada, ya que su dinámica no es inherente al sistema, sino impuesto del entorno.
- Las actividades se pueden describir usando diagramas de actividades.

Tareas

(Tareas)  
(Tareas)

# Diagrama de actividades (nivel intuitivo)

Actividad: Persona que entra en la sala de cine



asociados son las variables de entrada:

- tiempo entre llegadas  
(cuando la actividad llega)
- tiempo de servicio  
(cuando está atendido)

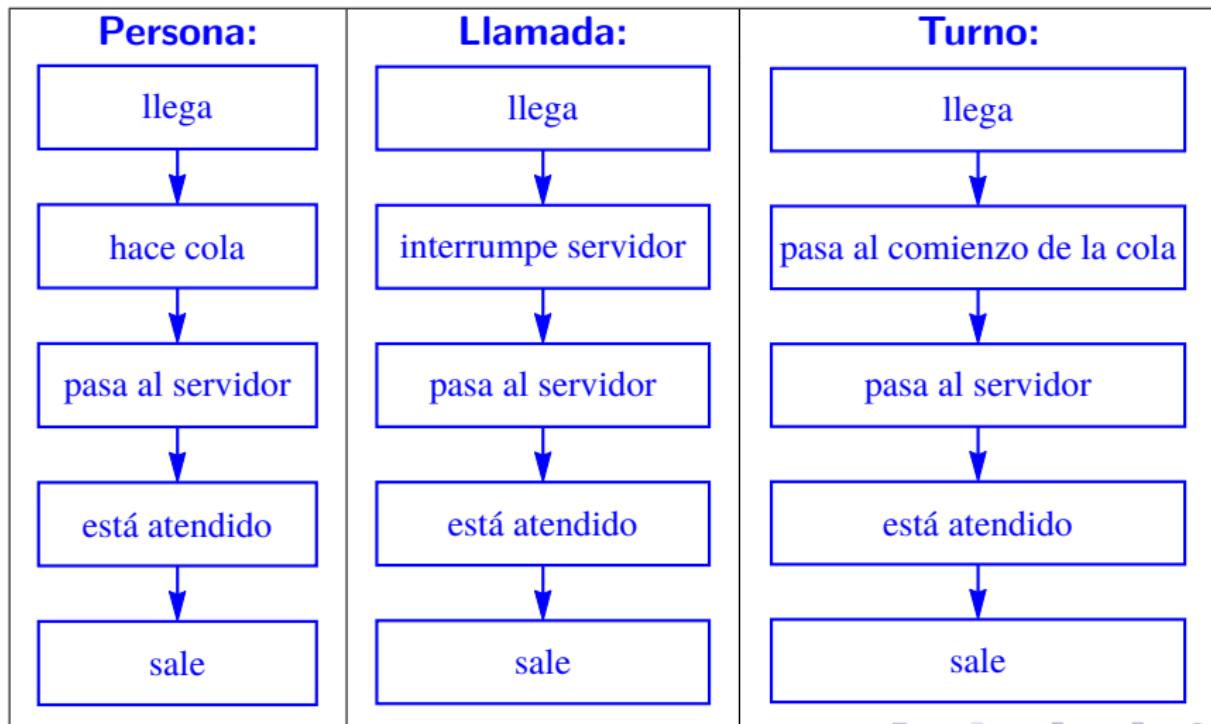
# Varias actividades en un modelo (1)

- Es posible que un modelo tenga varias actividades.
- En la simulación orientada a eventos, se pueden analizar los eventos correspondientes a cada actividad y mezclarlos durante la simulación

## Ejemplo:

- Al controlador de boletas llega una llamada al celular (Actividad: Llamada)  
Tiene asociado un tiempo entre llegadas y una duración
- Los controladores se turnan (Actividad: Cambio de turno)  
También tiene un tiempo entre llegadas y una duración

## Varias actividades en un modelo (2)



# Ejercicio en clases

## Especificar las actividades

### Problema del semáforo:

En una carretera de 2 carriles (uno en cada dirección) se está construyendo un puente, tal que un tramo de la carretera es de un solo carril. Se pone un semáforo en los dos lados de la construcción que tiene las fases rojo y verde: Si un lado está en verde, el otro obviamente debe estar en rojo, además se necesita un período de tiempo, donde ambos semáforos están en rojo para dejar pasar los carros que ya están en camino.



Variablos Entrada →

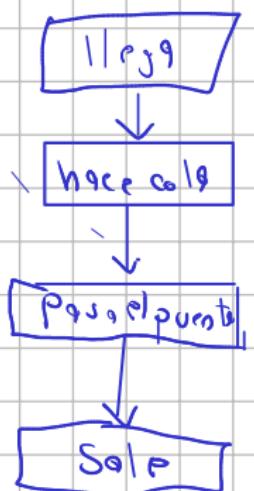
1) Tiempo Cambio semáforo

2) Contador Carrros

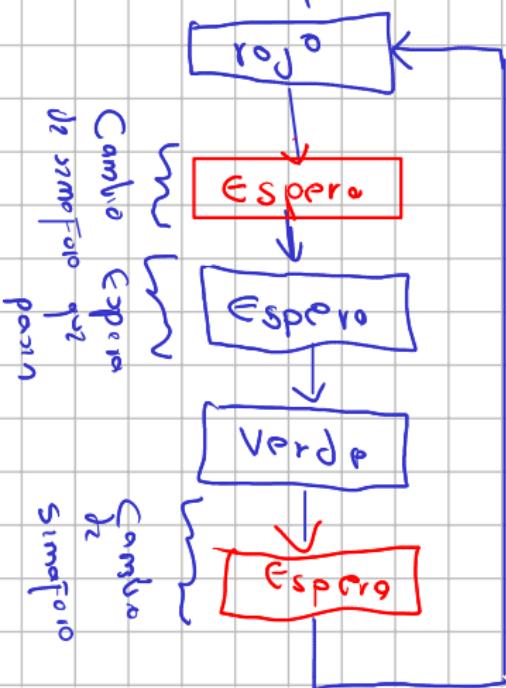
3) Tiempo Tienda pg. de carros UC →

4) Tiempo liberado carro

Carrro



Semáforo



# El papel de las actividades

- Las actividades son las entidades responsables para la dinámica del sistema y del modelo.
- Pueden existir varios tipos de actividades (como vimos), y no hay problema si hay dependencia entre las actividades (por ejemplo: la llamada interrumpe al atención de una persona, pero al finalizar la llamada, la atención se resume)
- La dinámica que las actividades transmiten al modelo se debe a las variables de entrada.
- Al especificar el diagrama de una actividad queda claro, donde y como se usarán las variables de entrada.
- Al pasar por el sistema, una actividad cambia el estado del sistema; afecta las variables de estado (dentro de un evento)

Variables de entrada: Son aquellas que se recogen del mundo real. Son las entradas (son datos conocidos).

Variables de estado: Son aquellas que permiten describir el sistema en un momento dado, supongan que es tomarle una foto en un instante dado a la simulación ¿Que la describe? - Tamaño actual de la cola, estado del servidor (cajero), numero de personas atendidas hasta el momento.

Variables de desempeño: Son los datos objetivo

# Variables de entrada

## Las variables de entrada

- se toman del sistema real (de una base de datos o se deben ser tomado explícitamente para el modelo)
- se modelan para conocer su naturaleza (se quiere conocer la distribución que tiene los datos, el "modelo de los datos") y
- se reproducen en el modelo de simulación para obtener datos aleatorios pero realistas (para cada variable de entrada se generan los aleatorios de acuerdo con la distribución encontrada)

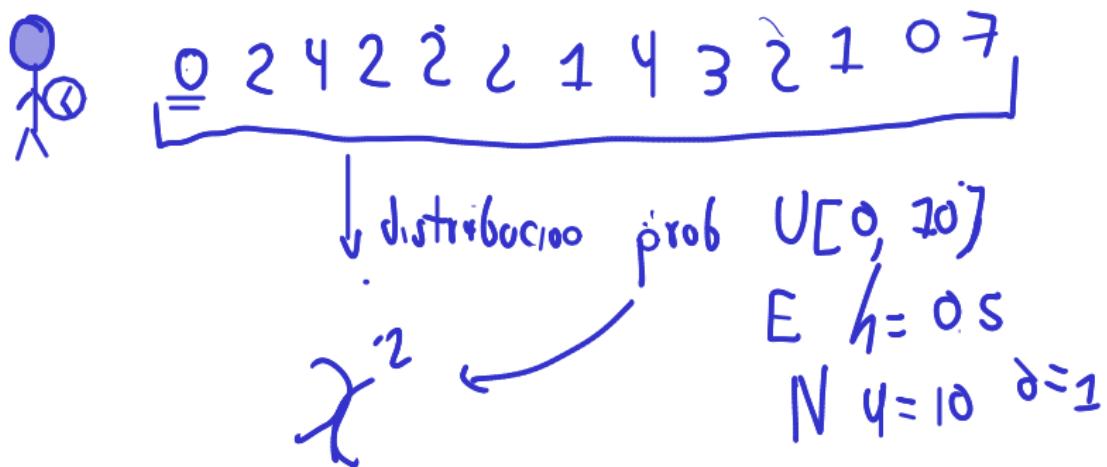
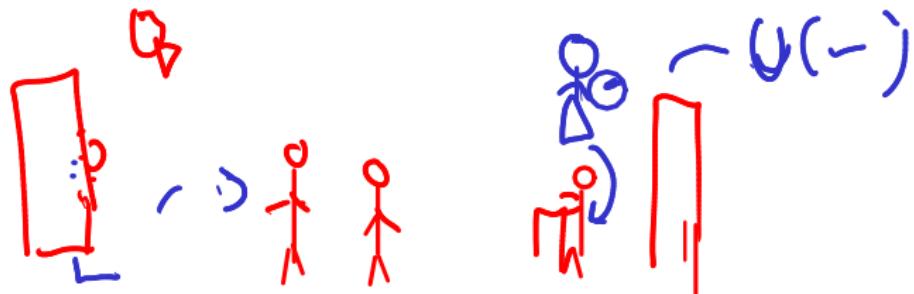
# El modelo de datos

## El proceso

### Variables de entrada.

- recolectar los datos
- proponer una distribución (histograma de frecuencias)
- estimar los parámetros de la distribución propuesta
- validar la distribución propuesta  
(por ejemplo por una prueba  $\chi^2$ )

Es un proceso iterativo: si la prueba termina en un rechazo, hay que buscar y proponer otra distribución.



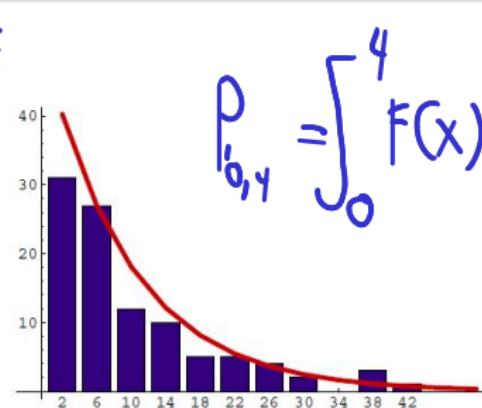
## Ejemplo 1: Tiempos entre llegadas (1)

- Se mide el tiempo cuando llegan 100 personas a una parada de buses.
- Dado que los tiempos de llegada son datos acumulados (la próxima persona llega más tarde que al anterior), se determina de estos datos el tiempo entre llegadas.
- Se construye el histograma de frecuencias  
(número de clases  $\approx \sqrt{\text{cantidad de datos}} = 10$ )

# Ejemplo 1: Tiempos entre llegadas (2)

Distribución propuesta

intervalo	frecuencia
[0,4)	31
[4, 8)	27
[8,12)	12
[12,16)	10
[16,20)	5
[20,24)	5
[24, 28)	4
[28,32)	2
[32, 36)	0
[36,40)	3
[40,44)	1



el histograma parece a una distribución exponencial, por eso:  
distribución propuesta  $\Rightarrow$   
distribución exponencial

## Ejemplo 1: Tiempos entre llegadas (3)

Estimación del parámetro  $\lambda$

El parámetro  $\lambda$  se relaciona con la media  $\mu$  de los datos que siguen esta distribución exponencial a través de

$$\mu = 1/\lambda$$

Entonces se puede estimar  $\lambda$  calculando el promedio de los datos  $\bar{x}$  y estimar

$$\lambda \approx 1/\bar{x}$$

# Ejemplo 1: Tiempos entre llegadas (4)

Estimación del parámetro  $\lambda$

intervalo	frecuencia
[0,4)	31
[4, 8)	27
[8,12)	12
[12,16)	10
[16,20)	5
[20,24)	5
[24, 28)	4
[28,32)	2
[32, 36)	0
[36,40)	3
[40,44)	1

$$\bar{x} = \sum_{\text{classes}} (\text{punto medio} \cdot \text{frecuencia}) / \text{número de datos}$$

$$\bar{x} = (2 \cdot 31 + 6 \cdot 27 + 10 \cdot 12 + \dots + 42 \cdot 1) / 100$$

$$\bar{x} = 10,04$$

$$\lambda \approx 1/\bar{x} = 0,1$$

## Ejemplo 1: Tiempos entre llegadas (5)

Prueba  $\chi^2$ : cálculo



intervalo	FO	FE	$(FE - FO)^2 / FE$
[0,4)	31	32.968	0.117
[4, 8)	27	22.099	1.087
[8,12)	12	14.813	0.534
[12,16)	10	9.93	0.0
[16,20)	5	6.656	0.412
[20,24)	5	4.462	0.065
[24, 28)	4	2.991	0.340
[28,32)	2	2.005	0.0
[32, 36)	0	1.344	1.344
[36,40)	3	0.901	4.89
[40,44)	1	0.604	0.26

La función de distribución de la exponencial es

$$F(x) = 1 - e^{-\lambda x}$$

Por eso un intervalo  $[a,b]$  tiene la probabilidad

$$\begin{aligned} \text{prob}([a, b]) &= F(b) - F(a) \\ &= 1 - e^{-\lambda b} - (1 - e^{-\lambda a}) \end{aligned}$$

$$\text{prob}([a, b]) = e^{-\lambda a} - e^{-\lambda b}$$

La frecuencia esperada es igual a la probabilidad por el número de datos

## Ejemplo 1: Tiempos entre llegadas (6)

Prueba  $\chi^2$ : conclusión

- El valor calculado de la  $\chi^2$  es 9.049
- Para el valor crítico de la  $\chi^2$  se debe tener en cuenta los grados de libertad.
- La fórmula general es:  
$$g.l = \text{numero de clases} - 1$$
  
- veces que se usó los datos para estimar parámetros
- en el caso específico:  $g.l. = 11 - 1 - 1 = 9$
- el valor crítico de la  $\chi^2$  para  $\alpha = 0.05$  es 16.92

**Conclusión:** dado que el valor calculado es menor que el valor crítico, se acepta la hipótesis que los datos tienen distribución exponencial con  $\lambda=0.1$

## Ejemplo 2: Tiempos de servicio (1)

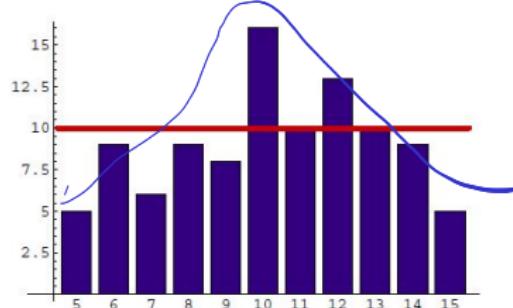
- Se mide el tiempo que dura la atención de 100 personas en una caja.
- Se construye el histograma de frecuencias  
(número de clases  $\approx \sqrt{\text{cantidad de datos}} = 10$ )

1. Simulación de eventos discretos
2. Actividades y variables de entrada
3. Eventos y variables de estado
4. Entidades y variables de desempeño
5. Conclusiones

## Ejemplo 2: Tiempos de servicio (2)

Distribución propuesta

intervalo	frecuencia
5	5
6	9
7	6
8	9
9	8
10	16
11	10
12	13
13	10
14	9
15	5



el histograma parece a una distribución uniforme, por eso:  
distribución propuesta  $\Rightarrow$   
distribución uniforme

## Ejemplo 2: Tiempos de servicio (3)

Estimación de los parámetros

Dado que los datos se obtiene en el intervalo [5,15], proponemos una distribución uniforme en [5,15].

Usamos los datos 2 veces: para estimar el punto inferior, y el punto superior.

1. Simulación de eventos discretos
2. Actividades y variables de entrada
3. Eventos y variables de estado
4. Entidades y variables de desempeño
5. Conclusiones

## Ejemplo 2: Tiempos de servicio (4)

Prueba  $\chi^2$

valor	$FO$	$FE$	$(FE - FO)^2 / FE$
5	5	9.091	1.841
6	9	9.091	0.001
7	6	9.091	1.051
8	9	9.091	0.001
9	8	9.091	0.131
10	16	9.091	5.251
11	10	9.091	0.091
12	13	9.091	1.681
13	10	9.091	0.091
14	9	9.091	0.001
15	5	9.091	1.841

La frecuencia esperada es  
 $100/11 = 9.091$

El valor calculado de la  $\chi^2$   
 es 11.981

$$g.l = 11 - 1 - 2 = 8$$

el valor crítico de la  $\chi^2$  con 8  
 g.l. para  $\alpha = 0.05$  es 15.51

**Conclusión:** se acepta la  
 hipótesis que los datos  
 tienen distribución  
 uniforme  $U[5,15]$

# Variables de estado

- Son las variables que muestran el estado del sistema en un momento dado (foto en este momento).
- Se cambian a través de las actividades.
- Los cambios del estado de un sistema definen los eventos de un sistema.
- Las variables de estado se tienen que tener en cuenta, para condicionar lo que pasa dentro de un evento.

# Variables de estado

## Ejemplo del cine

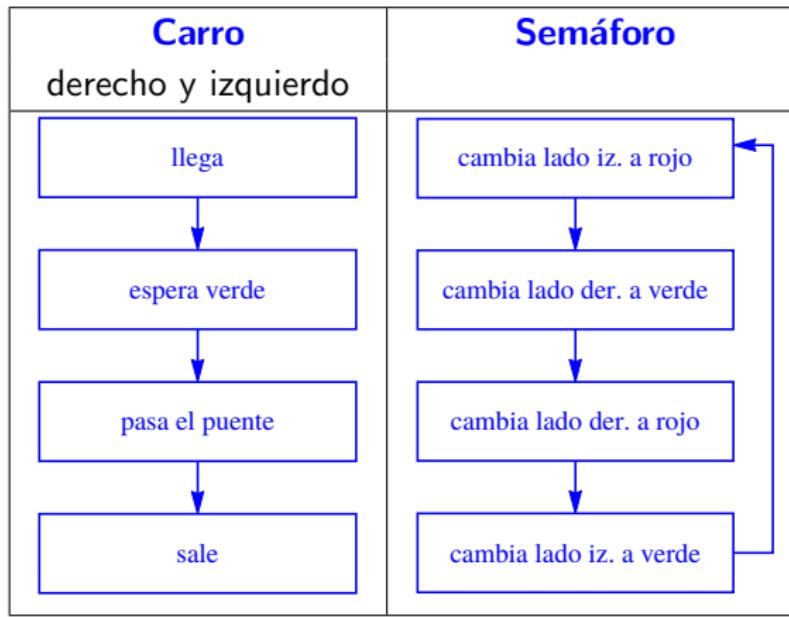
Hay dos variables de estado en este ejemplo:

- el número de personas esperando (tamaño de la cola)
- el estado del servidor (puede ser ocupado o desocupado)

## Ejercicio en clases

En el problema del semáforo: especificar las variables de estado

En el problema del semáforo, especificamos los siguientes actividades:



Carro

- Detrás / Avanzando
- Tiempo de la colg

Semáforo

- Rojo / Verde

# Eventos

- Los cambios de las variables de estado se asocian con eventos.
- Un evento ocurre en un momento determinado; no tiene duración
- Las ocurrencias de los eventos están provocados por las actividades
- Cada tipo actividad tiene su conjunto de tipos de eventos asociados
- Entre los eventos de una actividad se encuentran generalmente:
  - un evento de llegada, que ocurre cuando la actividad entra en el sistema
  - un evento de salida que ocurre cuando la actividad sale del sistema
- Eventos se pueden describir a través de diagramas de eventos.

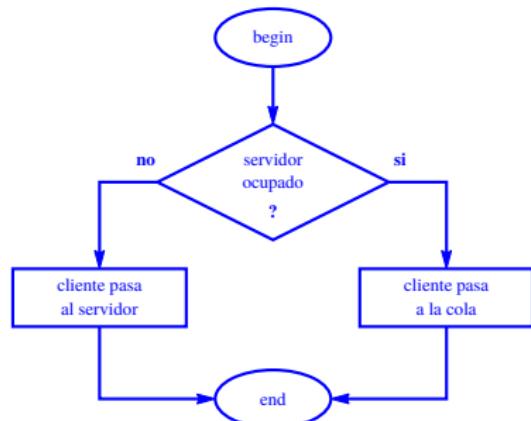
# Diagrama de eventos: Llegada (versión intuitiva)

Para un sistema con un servidor y una cola.

## Evento de llegada:

un cliente llega;  
si el servidor está ocupado,  
entra en la cola;  
sino ocupa al servidor.

No hay nada más que se puede hacer  
en este mismo instante.

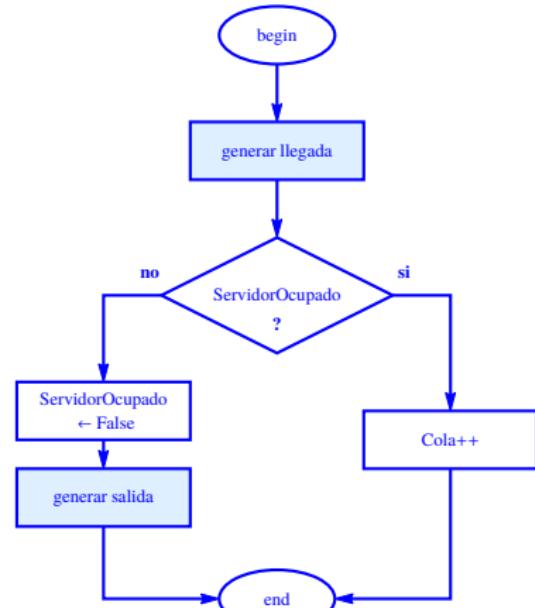


## Diagrama de eventos: Llegada (agregando la generación de nuevos eventos)

Se incluye la generación de una nueva llegada y una salida.

### Evento de llegada:

generar la llegada de próximo;  
si el servidor está ocupado,  
aumentar la cola;  
sino ocupar al servidor;  
generar salida de este cliente.

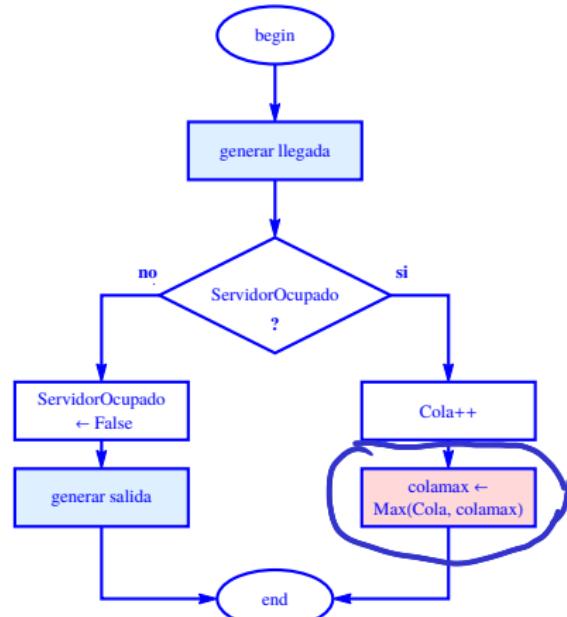


## Diagrama de eventos: Llegada (agregando variables de desempeño)

Se consideran las variables sencillas de desempeño: cola máxima y número de personas atendidas.

### Evento de llegada:

generar la llegada de próximo;  
si el servidor está ocupado,  
aumentar la cola;  
ajustar la variable cola máxima;  
sino ocupar al servidor;  
generar salida de este cliente.



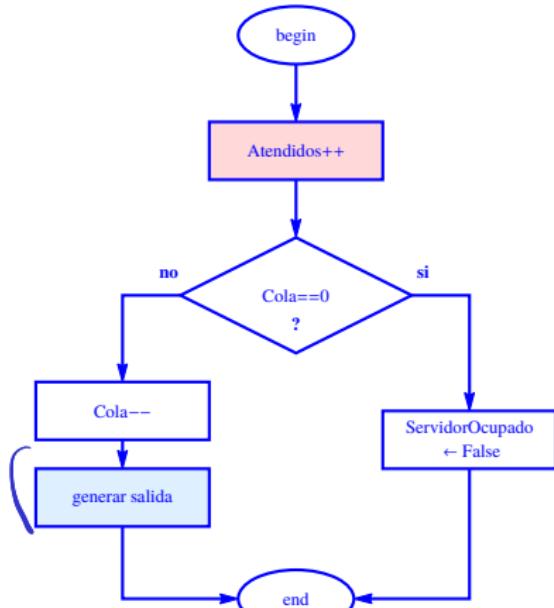
# Diagrama de eventos: Salida

Para un sistema con un servidor y una cola

Se consideran las variables sencillas de desempeño: cola máxima y número de personas atendidas.

## Evento de salida:

generar la llegada de próximo;  
si la cola es vacía,  
el servidor se pone desocupado;  
ajustar la variable cola máxima;  
sino reducir la cola;  
generar salida de este cliente.



# Ejercicio en clases

En el problema del semáforo: especificar los eventos

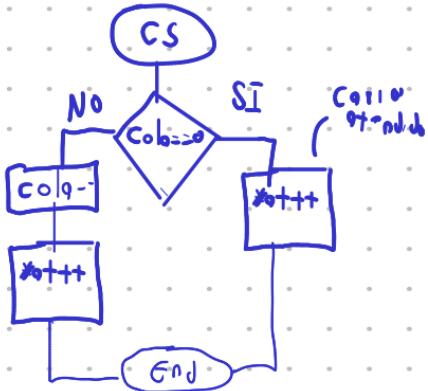
## Eventos identificados

- 1) Carro llega.
- 2) Carro sale
- 3) Semaforo cambia

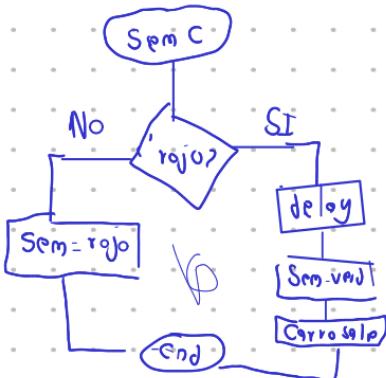
Carro llega



Carro Sale



Sem Cambia



# Eventos y variables de desempeño

## La respuesta están buscando de la simulación

Cuando el sistema cambia su estado, generalmente se ven afectados no sólo las variables de estado, sino también las variables de desempeño.

Entonces es en los eventos donde se actualizan también las variables de desempeño.

(Vimos como se hace esto con dos variables de desempeño muy sencillo, más adelante se consideran otras variables de desempeño típicas)

# Sincronización

El sistema simulado se cambia a través del tiempo.

Es decir, se deben actualizar las variables de estado en el momento en que ocurre uno de los eventos.

Para sincronizar, se debe organizar los eventos de acuerdo con su momento de ocurrencia.

Se necesita:

- un reloj del sistema para manejar el tiempo actual de la simulación
- una lista de eventos futuros (LEF) para almacenar los eventos futuros que se generan.

La generación de un evento es nada más que su programación para un momento futuro determinado, conocido cuando se genera.

## Reloj del sistema

- El reloj del sistema debe ser una variable entera para evitar que por errores de redondeo se cambia el orden lógico de la ocurrencia los eventos.
- Importante: definir adecuadamente la unidad del tiempo usado en la simulación. Unidades demasiado muy grandes pueden esconder características del desempeño del sistema.

En el ejemplo de la sala de cine se decide: la unidad del tiempo es un minuto.

## La lista de eventos futuros (LEF)

| (E, T)

- Un nodo de la lista consiste de la especificación del tipo de evento y su tiempo de ocurrencia (variable entera).
- La lista de eventos futuros se mantiene ordenada cronológicamente.
- En la simulación se ejecutan los eventos de acuerdo con su tiempo de ocurrencia desde la LEF.

Llegada: Va generar un evento de llegada (siguiente persona) y evento de salida (el mio)

## Generar eventos

La generación de un evento de un tipo determinado se refiere a la generación de su ocurrencia (frecuentemente a partir de una variable de entrada) y su inserción cronológica en la LEF.

### generar llegada:

- generar un aleatorio;
- convertirlo en un tiempo entre llegadas (de acuerdo al modelo de datos correspondiente);
- determinar la ocurrencia (oc) de la llegada, sumando el tiempo actual (reloj);
- insertar el nodo (L, oc) cronológicamente en la LEF.

Llegada

Salida

### generar salida:

- generar un aleatorio;
- convertirlo en un tiempo de atención (de acuerdo al modelo de datos correspondiente);
- determinar la ocurrencia de la salida, sumando el tiempo actual (reloj); → tomardo en cuenta el tiempo de S del reloj anterior
- insertar el nodo (S, oc) cronológicamente en la LEF.

# Ejecución de los eventos de acuerdo con el reloj del sistema

## Inicio:

- Se inicializa el reloj en 0.
- Se inicializan las variables de estado y de desempeño.
- Se inicializa LEF con un evento inicial de llegada (para cada actividad).

## Iteración:

- Se avanza el reloj hasta el tiempo de ocurrencia del próximo evento.
- Se realiza lo correspondiente al evento y lo saca de LEF.

## Parada:

- se itera hasta que se cumple una condición de parada, por ejemplo:
  - se llegó al total de tiempo de la simulación (reloj);
  - se llegó a simular un número predeterminado de personas;
  - la lista de eventos futuros está vacía.

# Ejecución de los eventos de acuerdo con el reloj del sistema

## Pseudocódigo

inicializar

while <not condición de parada>

    traer próximo elemento de la LEF

    ajustar reloj

    ejecutar el evento

fin\_while

    preparar información de desempeño

1. Simulación de eventos discretos
2. Actividades y variables de entrada
3. Eventos y variables de estado
4. Entidades y variables de desempeño
5. Conclusiones

# Ejemplo

Simulación manual de la entrada en la sala de cine usando los conceptos reloj y LEF

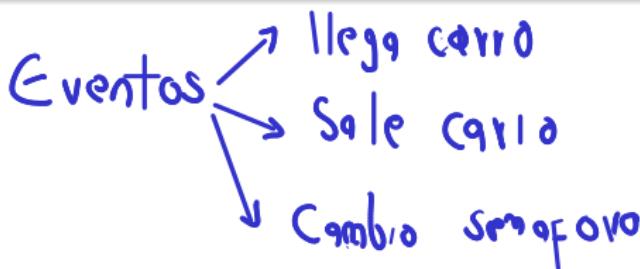
reloj	evento actual	atendidos	cola	colamax	serv_oc	generación llegada			generación salida			LEF
						random	tiempo entre llegadas	hora de llegada	random	tiempo de servicio	hora de salida	
		0	0	0	FALSE							$\{(0,L)\}$
0	(0,L)		0	0	TRUE	0.7126	39	39	0.3231	83		$\{(39,L), (83,S)\}$
39	(39,L)		1	1		0.4590	10	49				$\{(49,L), (83,S)\}$
49	(49,L)		2	2		0.4789	17	66				$\{(66,L), (83,S)\}$
66	(66,L)		3	3		0.9765	89	155				$\{(83,S), (155,L)\}$
83	(83,S)	1	2						0.0795	68	151	$\{(151,S), (155,L)\}$
151	(151,S)	2	1						0.0795	69	220	$\{(155,L), (220,S)\}$
155	(155,L)		2	3		0.0799	34	189				$\{(189,L), (220,S)\}$
189	(189,L)		3	3		0.2249	111	300				$\{(220,S), (300,L)\}$
220	(220,S)	3	2						0.1375	99	319	$\{(300,L), (319,S)\}$
300	(300,L)		3	3		0.2044	86	386				$\{(319,S), (386,L)\}$
319	(319,S)	4	2						0.7422	45	364	$\{(364,S), (386,L)\}$
364	(364,S)	5	1						0.2796	53	417	$\{(386,L), (417,S)\}$
386	(386,L)					0.9862	26	412				$\{(412,L), (417,S)\}$

## Observaciones

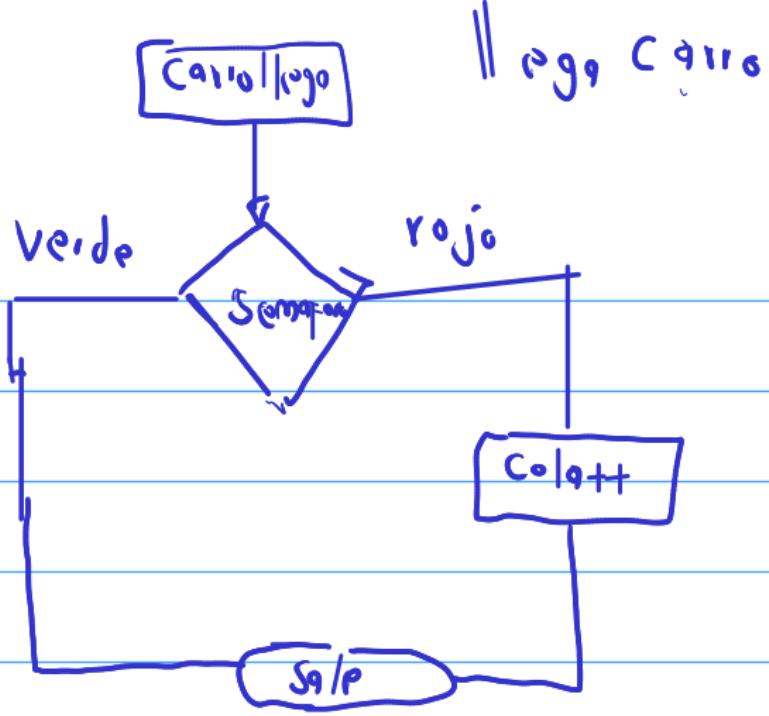
- Cada evento se ejecuta en el instante de su ocurrencia - no se mira ni el pasado ni el futuro (en la gráfica son los bloques coloreados en una sola línea).
- El reloj se ajusta a la ocurrencia del evento más próximo.
- Cada evento puede generar otros eventos.
- Cada evento puede modificar las variables de estado (servidor ocupado, cola).
- Cada evento puede modificar las variables de desempeño (atendidos, colamax).
- Los aleatorios se generan cuando se los necesitan (se gasta poca memoria).
- La lista de eventos futuros LEF es corta (es decir, su manejo es eficiente).

## Ejercicio en clase

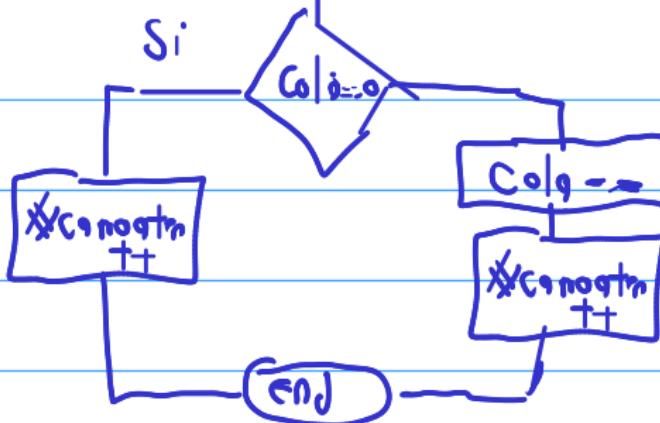
Simulación manual de la entrada en la sala de cine con 2 puertas



llega carro



Cerro Slip



Semáforo Cambia



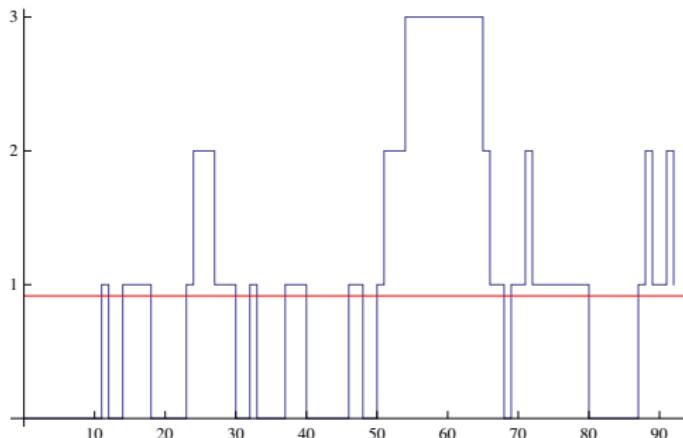
## Variables de desempeño

- Variables de desempeño miden el comportamiento del sistema
- Están relacionados estrechamente con el problema (ej.: no es importante medir el comportamiento de la cola de entrada a las urgencias de un hospital, si el problema es el suministro de materiales)
- Miden como se comportan las diferentes entidades del sistema/modelo
- Dado que las variables de estado son variables aleatorias, las variables de desempeño se expresan con frecuencia como promedios, porcentajes o valores esperados.
- Para una descripción realista del sistema/modelo se requiere evaluarlo varias veces (varios turnos, días, meses según el caso) o durante un tiempo bastante largo (lo que no es siempre posible)

# Comportamiento del sistema/modelo

## Sistema en equilibrio

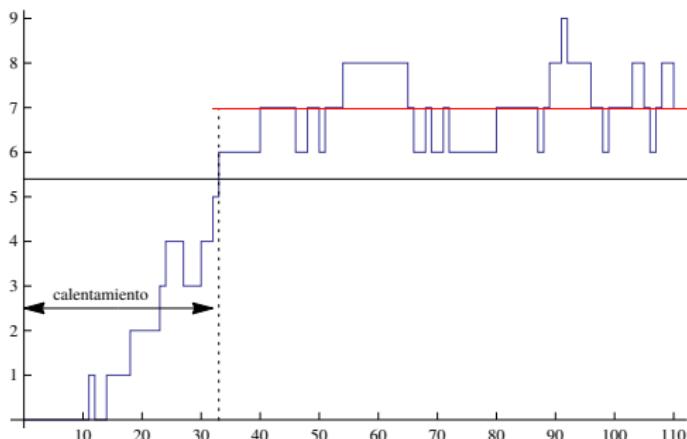
- El sistema se encuentra en equilibrio.
- La variable de estado oscila alrededor de un valor constante (en rojo)
- Este valor es el valor esperado de esta variable



# Comportamiento del sistema/modelo

Sistema en equilibrio después de un calentamiento

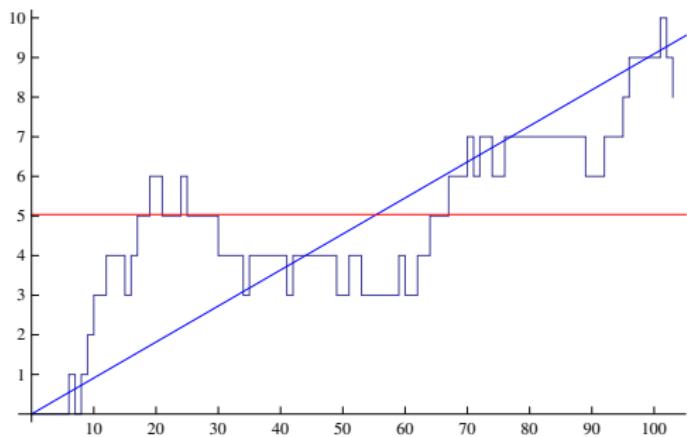
- El sistema alcanza el equilibrio después de un período de calentamiento
- El valor esperado se ve afectado por el calentamiento
- Para evitar este sesgo, se toma el valor esperado después del calentamiento (en rojo)



# Comportamiento del sistema/modelo

Sistema fuera del equilibrio

- El sistema no se encuentra en equilibrio
- Tiene tendencia creciente (en azul)
- El valor esperado (en rojo) no da sentido en este caso



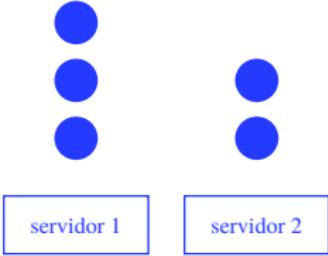
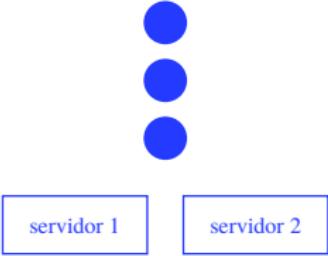
# Entidades

- Colas
- Servidores
- Multiservidores
- Actividades

1. Simulación de eventos discretos
2. Actividades y variables de entrada
3. Eventos y variables de estado
4. Entidades y variables de desempeño
5. Conclusiones

# Entidad: Cola

Tipos de sistema de cola-servidor

en línea	en paralelo	una cola, varios servidores
 <p>servidor 1</p>	 <p>servidor 1</p> <p>servidor 2</p>	 <p>servidor 1</p> <p>servidor 2</p>

# Entidad: Cola

## Características

procesos asociados	variables de estado asociadas	variables de desempeño asociadas
entrar en cola; salir de la cola;	tamaño de la cola;	cola máxima; tiempo promedio de espera en cola; valor esperado del tamaño de la cola.

## Entidad: Cola

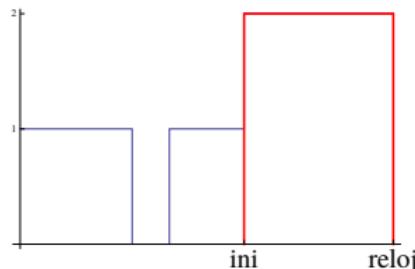
### Cálculo del valor esperado de la cola (1)

Cada vez que se cambia la cola, se llama antes del cambio un procedimiento (aquí: "*ColaAcum*") que calcula el valor del último rectángulo y lo acumula a un acumulador ("*tamcola*").

En una variable "*ini*" se guarda el tiempo del cambio anterior.

*ColaAcum* :

```
tamcola += (reloj - ini) * cola;  
ini ← reloj
```



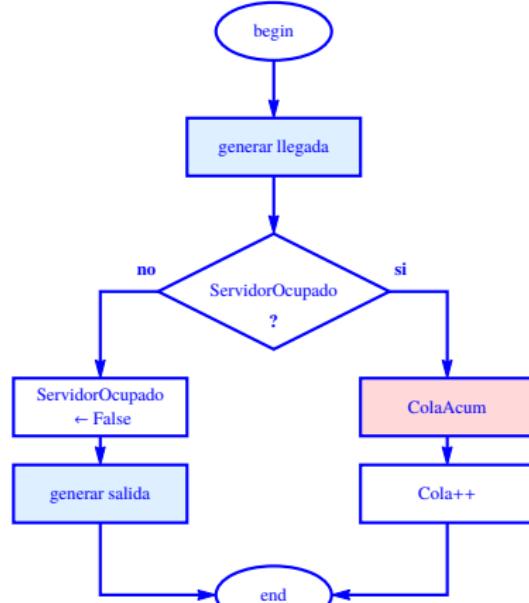
Finalizado la simulación se debe determinar el valor esperado por:  
 $tamcola/reloj$

1. Simulación de eventos discretos
2. Actividades y variables de entrada
3. Eventos y variables de estado
4. Entidades y variables de desempeño
5. Conclusiones

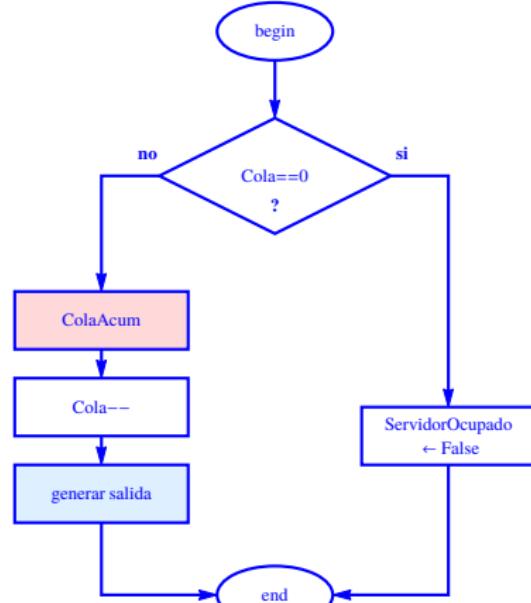
# Entidad: Cola

Cálculo del valor esperado de la cola (2)

## Evento de llegada



## Evento de salida



## Entidad: Cola

Cálculo del tiempo promedio de espera en cola

De pronto la forma más fácil es implementar la cola como una lista, que guarda los tiempos de llegada.

**Adicionar** un elemento a la cola corresponde a agregar el nuevo tiempo de llegada al final de esta lista

Antes de **quitar** un elemento de cola, se calcula el tiempo de espera (*reloj – tiempo de llegada*) y se acumulan estos tiempos.

Finalizado la simulación se determina el tiempo de espera **promedio**, dividiendo por el total de atendidos.

Adicionar y quitar elementos de la cola se hace dentro de los eventos de llegada o salida respectivamente.

# Entidad: Cola

Uso para determinar el tiempo entre dos procesos de una actividad

Se usa una cola ficticia.

Comenzando el primero de los procesos, se entra la actividad en una cola.

En el segundo proceso, la actividad sale de la cola ficticia.

Implementando el cálculo del tiempo promedio de espera como arriba, se obtienen el tiempo entre los dos proceso.

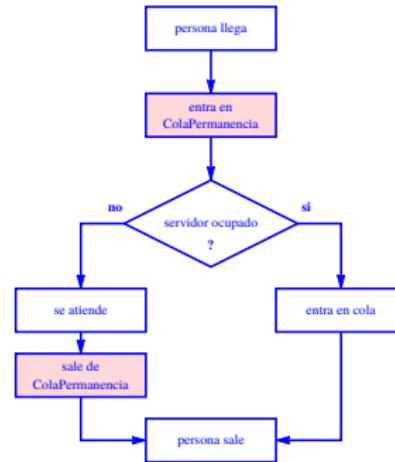
Adicionar y quitar elementos de la cola se hace dentro de los eventos de llegada o salida respectivamente.

## Entidad: Cola

Uso para determinar el tiempo entre dos procesos de una actividad: Ejemplo

Si queremos determinar el tiempo de permanencia en el sistema de la sala de cine, podemos introducir la cola ficticia (*colaPermanencia*). El diagrama de actividades de la persona que va a cine se modifica correspondientemente. Usar el cálculo del valor esperado del tiempo en cola para esta cola ficticia, nos da el valor esperado de la permanencia en el sistema.

### Diagrama de la actividad "persona":



# Entidad: Servidor

## Características

procesos asociados	variables de estado asociadas	variables de desempeño asociadas
ocupar / desocupar servidor; interrumpir / reanudar;	ocupado/ desocupado;	número de personas atendidas;  % de utilización del servidor.

# Entidad: Servidor

## Interrumpir y reanudar

- La actividad que interrumpe el servidor tiene sus eventos de llegada y salida (aquí los llamamos LlegadaAI y salidaAI).
- En el evento de llegada interrumpe el servidor (si está ocupado) y es responsable para reprogramar la actividad que en este momento está ocupando el servidor. La reprogramación debe aplazar la salida de la actividad que ocupa el servidor por el tiempo de atención de AI. Este valor se debe guardar.
- Cuando en su evento de salida está liberando el servidor, la actividad retoma el servidor y continua. Cuando llega su salida hay que programar la salida nueva teniendo en cuenta el tiempo que se aplazó su terminación.

# Entidad: Multiservidor

## Características

Un multiservidor se puede usar cuando se quiere modelar varias servidores con características iguales (tiempos de servicio, comportamiento) que procesan una cola común.

Un multiservidor tiene una capacidad máxima, un parámetro fijo del multiservidor.

Tiene asociada la capacidad ocupada, que es la variable de estado que mide, cuantas unidades del multiservidor están ocupadas.

procesos asociados	variables de estado asociadas	variables de desempeño asociadas
ocupar desocupar capacidad del multiservidor;	capacidad ocupada;	número de personas atendidas; valor esperado de la capacidad utilizada.



# Entidad: Multiservidor

Valor esperado de la capacidad utilizada

La capacidad utilizada es una variable aleatoria que toma valores enteros entre 0 y la capacidad máxima del multiservidor. El valor esperado tiene en cuenta por cuanto tiempo se mantiene un valor específico.

La capacidad utilizada es una variable aleatoria con características muy similares al tamaño de una cola, por eso, el cálculo de su valor esperado se hace en forma análoga.

# Entidad: Multiservidor

Uso como una bodega

Las características del multiservidor permiten usarlo para simular una bodega.

- La bodega tiene una capacidad máxima
- Ítemes pueden entrar en la bodega, ocupando capacidad
- Ítemes salen de la bodega, liberando capacidad

El valor esperado de la capacidad utilizado dice que tan llena está la bodega en el promedio.

## Conclusiones (1)

- El motor de la simulación de eventos discretos está dado por la interacción entre Lista de Eventos Futuros LEF y el reloj del sistema.
- Por esta interacción está resuelta la sincronización de los eventos, puesto que la LEF es una lista con los tipos de eventos futuros y si momento de ocurrencia.
- Los eventos están definidos a través de las actividades que provocan cambios en las variables del estado del sistema. Los diagramas de actividades y de eventos ayudan en su implementación.
- Cada evento realiza los cambios en el mismo instante de su ocurrencia - no tiene duración.
- Es importante considerar dentro de los eventos la generación de los futuros eventos pertinentes, porque si la lista de eventos

## Conclusiones (2)

Hay 3 tipos de variables aleatorias en un modelo de simulación: Las variables de entrada, las variables de estado y las variables de desempeño.

Todas estas variables se usan y/o modifican dentro de los eventos.

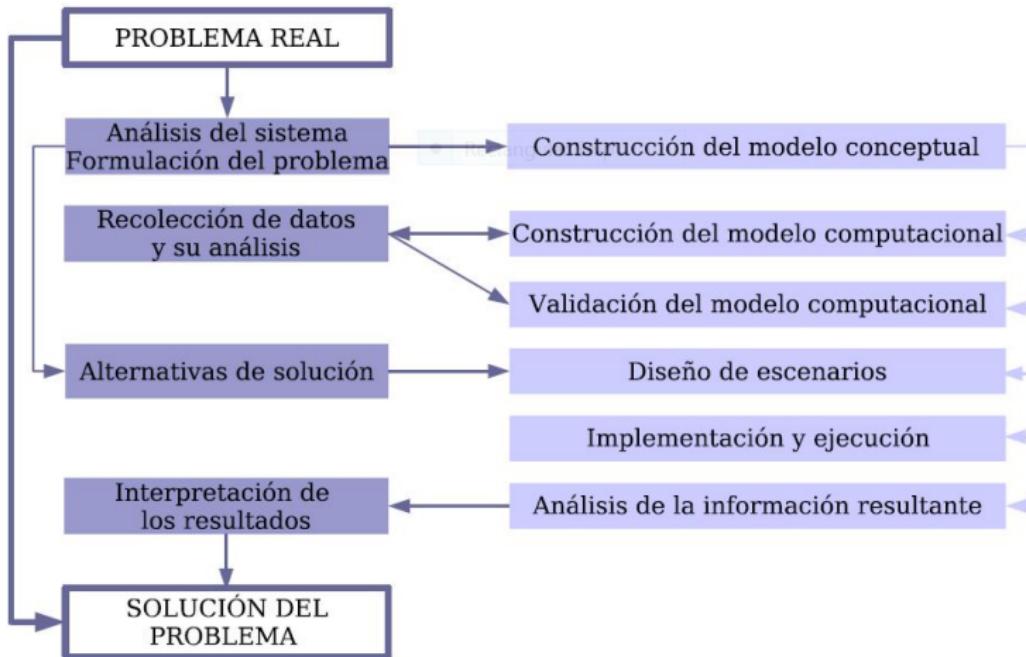
1. Las variables de entrada se modelan con los datos del sistema real. Son responsables para dar dinámica al sistema simulado.
2. Las variables de estado son necesarias para describir y cambiar el estado del sistema. Se modifican dentro de los eventos, cuando el sistema sufre un cambio provocado por el evento.

## Conclusiones (3)

3. Las variables de desempeño se usan para determinar que tan bueno funciona el sistema. Está orientado en el problema que se trata de resolver con el modelo de simulación. Tienen 2 funcionalidades importantes dentro de la simulación:
  - Sirven para comparar el desempeño del sistema simulado con el sistema real, si no tienen el mismo comportamiento, hay que revisar el modelo de simulación hasta encontrar equivalencia entre sistema real y simulado.
  - Sirven para medir la bondad de las alternativas propuestas para el mejoramiento del sistema, implementadas en los diferentes escenarios.

Finalizando la simulación, se tiene que aplicar generalmente un proceso para completar el cálculo de las variables de desempeño

# Pasos de un estudio de simulación



# Problema real / Análisis del sistema (1)

- Simulación se hace sólo si se quiere resolver un problema real (soporte a las decisiones)
- Precondición es el buen conocimiento del problema
- Por eso es importante el enfoque de sistemas
- Los componentes, el funcionamiento y los datos se obtienen del sistema real
- Las propuestas de solución deben también ser viables en el sistema real



# Problema real / Análisis del sistema (2)

## Información para las variables de entrada

- Se debe construir el modelo de datos.
- Luego se construye el modelo de datos que permite determinar la distribución que tienen los datos originales.
- En el modelo de simulación se generan datos con la misma distribución.

## Información para las variables de desempeño

- Las variables de desempeño son importantes para evaluar el comportamiento de sistemas
- Deben ser deducidos en el sistema real para tener la base de comparación para
  - la validación del modelo;
  - la comparación con el comportamiento en los escenarios.

# Modelo conceptual (1)

- El modelo trata a captar los factores importantes del problema a modelar
- Importante es encontrar el nivel adecuado entre un modelo sencillo y la complejidad requerida para reflejar el comportamiento del sistema real
  - demasiado sencillo: el modelo no describe el problema correctamente, la evaluación de las soluciones no es confiable
  - demasiado complejo: la implementación y el análisis se vuelven demasiado complejos.

## Modelo conceptual (2)

En el modelo conceptual se decide:

### Sistema/Problema:

- ¿cuáles componentes se deben modelar?;
- determinar el grado de detalle del funcionamiento a incluir;
- tener en cuenta las posibles alternativas de solución (para preverlos en el modelo computacional);
- ¿cuáles son las variables de entrada?, ¿son disponibles?, ¿requieren preprocesamiento?;
- ¿cuáles variables de desempeño son relevantes para el problema? (no es la cantidad de variables que cuenta; se deben seleccionar aquellas que son relevante para el problema).

## Modelo conceptual (3)

En el modelo conceptual se decide:

### **Modelo:**

- ¿qué variables de estado se deben considerar en acuerdo de la complejidad del modelo?
- ¿cuáles entidades se deben modelar?;
- ¿cuáles son las actividades del modelo?, ¿cómo se mueven por el sistema?
- definir los eventos que cambian el estado del sistema; pensar también en las variables de desempeño;
- ¿cómo afectan las alternativas de solución a los escenarios?, ¿cuales son los cambios correspondientes en el modelo?

# Modelo computacional (1)

- Uso de hojas electrónicas (problemas sencillo)
- uso de lenguajes de alto nivel para desarrolla modelos de simulación orientada a eventos (problemas complejos)
- uso de software de simulación (GPSS, Promodel, Arena, SimFactory,...)

# Modelo computacional (2)

## Tener en cuenta

- seleccionar el generador de aleatorios
- determinar la unidad de tiempo (todos los tiempos deben ser enteros para evitar un posible error de la sincronización por errores de redondeo (que no ocurren en datos enteros))
- desarrollar los modelos de datos para las variables de entrada

## Modelo computacional (3)

Ayuda para el desarrollo:

- diagramas de actividades
- diagramas de eventos
- estructura fija del manejo de la simulación a través de LEF y reloj

# Importancia de la validación

- Se construye un modelo que debe ser suficiente cercano al sistema problema para poder concluir del modelo al sistema.
- Sólo si se tiene seguridad que modelo y sistema coinciden en sus rasgos importantes, se puede tener confianza en las conclusiones.
- Por eso es esencial realizar la validación



# Aspectos a validar (1)

Validación de apariencia:

- revisar el realismo del modelo
- estructura del sistema
- comportamiento en situaciones normales
- comportamiento en situaciones extremas
- confiabilidad de los datos

Validación de supuestos:

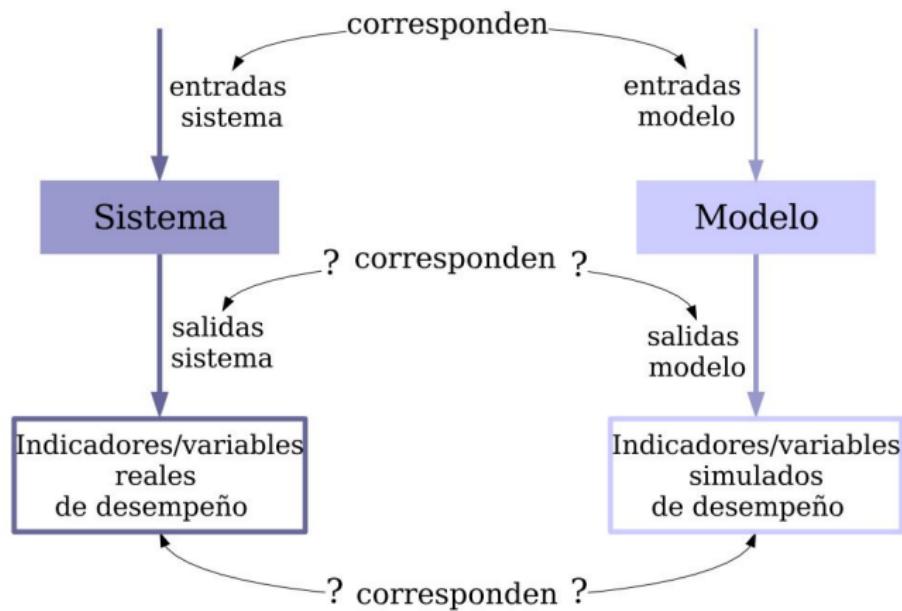
- contrastar sistema y modelo
- verificar el impacto de las simplificaciones que hace el modelo
- validar los modelos de datos

## Aspectos a validar (2)

Validación de transformaciones input-output:  
comparar el output del sistema con el output del modelo para un input similar.

- el comportamiento de las variables de estado debe ser similar
- el comportamiento de las variables de desempeño debe ser similar

# Validación de las transformaciones input-output (1)



## Validación de las transformaciones input-output (2)

- Las variables de estado y de desempeño de un sistema son generalmente variables aleatorias.
- Por eso, procedimientos estadísticas son los más adecuados para validarlas.
- Generalmente se usan pruebas de hipótesis para mostrar que las variables en el sistema real y las variables correspondiente resultando de la simulación tienen el mismo comportamiento.

# Heurísticas para la validación (1)

- Comparar promedios y variaciones de las variables entre sistema real y modelo.
- Compara las tendencias de las variables de estado.
- Construir intervalos de confianza para las variables reales y simuladas y compararlos.

## Heurísticas para la validación (2)

- Pruebas con datos históricos del sistema real: En vez de generar aleatoriamente los datos en la simulación, se usa un conjuntos de datos obtenidos del sistema real. Se espera que los indicadores de desempeño del sistema simulado estén cerca a los obtenidos de los datos históricos directamente.
- Prueba de Turing: Se preparan 10 reportes de salida idénticos, 5 obtenidos del sistema real, otros 5 resultados de la simulación. Se mezclan bien estos reportes y pide a una persona experta del sistema real que identifique los reportes reales y simulados. Si el experto está capaz de identificar la mayoría de ellos, el modelo NO describe bien el sistema. Si el experto no nota la diferencia, el modelo es válido.

## Definir alternativas de solución (1)

Si el modelo está validado, se puede usarlo para realizar experimentos (experimentación simulada).

La definición de alternativas de solución es una actividad donde deben participar expertos que conocen bien el sistema.



## Definir alternativas de solución (2)

Entre los factores a considerar en la definición de alternativas de solución se tiene:

- la factibilidad de su implementación en el sistema real
- el costo asociado y tiempo requerido para realizar los cambios
- la disposición al cambio de las personas afectadas
- la dificultad de transición al sistema modificado

# Ejemplo de un problema cola-servidor sencillo

Problemas que se presentan

- La cola está creciente a lo largo de tiempo
- La cola se vuelve muy larga
- El servidor está ocupado casi todo el tiempo

Esto parece indicar que la demanda para el servicio es alta en comparación con la capacidad del sistema. Se sospecha que se pierden clientes que no están dispuestos de esperar. Se buscan alternativas que permiten atender los clientes más eficientemente.

# Ejemplo de un problema cola-servidor sencillo

## Definir alternativas de solución

Se plantean 3 alternativas:

- ① Reducir el tiempo de atención del servidor (ej.: entrenamiento, ayudas tecnológicas, un ayudante). Se espera que servidor puede bajar los tiempos de atención
- ② Partir el proceso del servidor en 2 etapas en línea (por ej.: servidor1: atiende clientes y vende, servidor2: recibe pago y empaca). Esto puede bajar el tiempo de atención del servidor1, pero requiere un segundo servidor. Se estima que servidor ahorra el tiempo de la segunda etapa, que realiza servidor2.
- ③ Instalar un segundo servidor. Se supone que el segundo servidor trabaja con la misma velocidad que el primero.

# Ejemplo de un problema cola-servidor sencillo

## Diseñar escenarios

Acorde con las alternativas de solución se plantean los siguientes escenarios de simulación:

Escenario 0: la situación actual

Escenario 1: servidor con ayudante

Escenario 2: proceso en línea

Escenario 3: dos servidores

OJO: es importante tener estimaciones realistas de cómo se cambian los datos en los diferentes escenarios.

# Experimentación computacional

- Los escenarios son realistas
- Se implementan, tratando cambiar lo menos posible el modelo correspondiente al escenario 0, ya que éste está validado
- Los escenarios son experimentos que se realizan en el computador, por eso se deben aplicar conceptos estadísticos relacionados (tamaño de la muestra, evaluación de los resultados)
- Ventaja de la experimentación en el computador: las repeticiones son mucho más económicas; experimentos reales consuman generalmente mucho dinero y tiempo

# Implementación de los escenarios y Análisis de los resultados

- Se define el número de simulaciones en cada escenario.
- Se hacen las modificaciones pertinentes del modelo para adaptarlo a cada uno de los escenarios.
- Se simula, guardando los valores de los indicadores seleccionados.
- Se analiza cada escenario y lo compara con el escenario 0 (promedio y variaciones, comportamientos, intervalos de confianza).

Ver Simulación en Mathematica para el procedimiento completo

## Conclusiones

- Antes de desarrollar el modelo se debe conocer bien el sistema real y el problema que se presenta
- El modelo implementado debe ser validado cuidadosamente, probablemente en un proceso iterativo, donde se corrige y complementa el modelo
- Una vez validada el modelo, se lleva cabo la experimentación computacional: definiendo, ejecutando y analizando escenarios que corresponde a alternativas realistas de solución
- Comparando los escenarios se puede proponer soluciones al problema real.