

# Modelado y Simulación de Eventos Discretos

Materia: Simulación de procesos

Prof. Dr. BARSEKH-ONJI Aboud

Facultad de Ingeniería  
Universidad Anáhuac México

January 27, 2026

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Sistemas Continuos vs. Discretos

## Sistemas Continuos

- Los cambios son predominantemente **suaves** e ininterrumpidos.
- **Ejemplo:** El movimiento de una aeronave.

## Sistemas Discretos

- Los cambios son predominantemente **discontinuos** y ocurren en puntos específicos en el tiempo.
- **Ejemplo:** Una fábrica donde la finalización de un producto es un evento discreto.

# Ambigüedad en la Representación

La **descripción** de un sistema, más que su naturaleza inherente, determina qué tipo de modelo se utilizará. El propósito del modelo y el nivel de detalle son cruciales.

# Discrete Event Systems (DES)

## Definición

Un **Sistema de Eventos Discretos (DES)** es un sistema dinámico que evoluciona de acuerdo con la ocurrencia **asíncrona** de eventos físicos.

- Su dinámica se caracteriza por la ocurrencia de eventos discretos.
- Requieren control y coordinación para asegurar el flujo ordenado de eventos.
- Ejemplos: sistemas de manufactura automatizados, redes de comunicación, sistemas de software.

# Características Clave del Modelado DES

## ¿Cómo se diferencia de los modelos tradicionales?

Los modelos de eventos discretos se distinguen por:

- **Paso del tiempo:** Es **impulsado por eventos** (event-driven), no por pasos de tiempo fijos. El modelo "salta" de un evento al siguiente.
- **Coordinación:** Los componentes son **asíncronos** y débilmente coordinados, lo que permite una representación más realista.
- **Eficiencia:** La simulación es inherentemente eficiente, ya que concentra el procesamiento solo en los eventos, que son cambios significativos y relativamente raros.

# Uso del Modelado y Simulación DES

## Predicción y Evaluación

El modelado de eventos discretos es un procedimiento matemático para describir un proceso dinámico. La simulación del modelo permite predecir posibles situaciones para evaluar y mejorar el rendimiento del sistema.

# Uso del Modelado y Simulación DES

## Análisis "What-If" (¿Qué pasaría si...?)

Se utiliza para crear predicciones de los estados del sistema, que pueden modificarse para examinar situaciones hipotéticas.

- **Ejemplo clásico:** Evaluar una línea de espera (cola).
- **Preguntas a responder:** ¿Cuánto tiempo esperará un cliente en promedio?  
¿Cuántos servidores se necesitan para reducir la espera?



# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Definiciones Clave (Parte 1) I

## Entidad (*Entity*)

Un objeto de interés en el sistema. Su selección depende del propósito y nivel de abstracción del estudio.

## Atributo (*Attribute*)

Propiedades que describen a las entidades. Son valores locales que pertenecen a una entidad específica (e.g., tiempo de llegada, color de una pieza).

## Definiciones Clave (Parte 1) II

### Estado (*State*)

Un conjunto de variables que caracteriza completamente al sistema en cualquier momento.

### Evento (*Event*)

Una incidencia instantánea que podría resultar en un cambio de estado. Puede ser:

- **Endógeno:** Generado por el propio sistema.
- **Exógeno:** Inducido por el entorno del sistema.

# Entidades, Atributos y Actividades I

Sistema	Entidades	Atributos	Actividades
Mercado	Clientes	Artículos de compra	Pagar en caja
Sistema telefónico	Mensajes	Longitud	Transmisión
Banco	Cliente	Estado del saldo	Depositar o retirar
Tráfico	Automóviles	Velocidad y distancia	Conducir
Control de calidad	Productos	Calidad	Comprobar
Sistema de fábrica	Productos	Pedidos pendientes	Llegada de pedidos

Table 1: Ejemplos para diferentes sistemas.

# Definiciones Clave (Parte 2) I

## Servidor (*Server*)

Una entidad que proporciona servicio a entidades dinámicas (e.g., servidores, máquinas). Las entidades pueden tener que esperar en una **cola** (*queue*) si el recurso está ocupado.

## Procesamiento de Listas (*List processing*)

Las colas son listas de entidades que esperan. Se pueden procesar con diferentes lógicas: FIFO (primero en entrar, primero en salir), LIFO (último en entrar, primero en salir) o por prioridad según un atributo (e.g., tiempo de procesamiento más corto).

## Definiciones Clave (Parte 2) II

### Actividades y Retrasos (*Activities and delays*)

- Una **actividad** es una duración de tiempo conocida de antemano (e.g., un tiempo de servicio).
- Un **retraso** es una duración de tiempo indefinida causada por las condiciones del sistema (e.g., el tiempo de espera en una cola).

## Definiciones Clave (Parte 2) III

### Actividades Exógenas vs. Endógenas

- *Endógenas*: Ocurren dentro del sistema.
- *Exógenas*: Ocurren en el entorno y afectan al sistema. Un sistema sin actividades exógenas es un sistema **cerrado**.

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 **Sistemas de Colas (*Queuing Systems*)**
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización



# Sistemas de Colas (*Queuing Systems*)

Una aplicación importante de la simulación de eventos discretos es el estudio de la dinámica de las líneas de espera o **colas**.

# Razones para estudiar las colas

- Determinar el uso óptimo de los servidores.
- Analizar el tiempo de espera de los clientes.
- Decidir el espacio de espera requerido.
- Contar el número de clientes atendidos

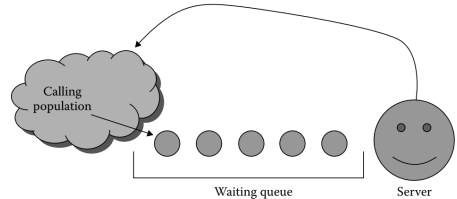


Figure 1: Sistema de Colas.

# Componentes de un Sistema de Colas I

**Población de Llamada (*Calling Population*):** Es la población de donde provienen los clientes (e.g., personas, máquinas, pedidos). Puede ser finita o infinita.

**Capacidad del Sistema (*System Capacity*):** Es el número máximo de clientes que puede haber en el sistema (en la cola + en servicio). Puede ser limitada o ilimitada.

**Proceso de Llegada (*Arrival Process*):** Caracteriza los tiempos entre llegadas de clientes sucesivos. Puede ser:

- **Programado:** Citas médicas, vuelos programados.
- **Aleatorio:** Clientes en un banco, pedidos a una fábrica.

# Comportamiento y Rendimiento

## Comportamiento en la Cola

Se refiere a la acción del cliente en la cola:

- **Balk:** Abandona al ver que la línea es demasiado larga.
- **Reneg:** Abandona después de un tiempo al ver que la línea avanza muy lento.
- **Jockey:** Se mueve de una línea a otra buscando la más corta.

## Medidas de Rendimiento

Indicadores clave del sistema:

- Tiempo promedio en el sistema.
- Utilización del servidor.
- Tiempo de espera por cliente.
- Número de clientes atendidos.

# Tipos de Simulación para Sistemas de Colas

## Simulación Orientada al Tiempo Discreto

- Se refiere a la simulación en intervalos de tiempo **iguales** y fijos.
- El reloj avanza en incrementos constantes ( $\Delta t$ ).
- En cada paso, se revisa si ha ocurrido un evento.

# Tipos de Simulación para Sistemas de Colas

## Simulación Orientada al Tiempo Discreto

- Se refiere a la simulación en intervalos de tiempo **iguales** y fijos.
- El reloj avanza en incrementos constantes ( $\Delta t$ ).
- En cada paso, se revisa si ha ocurrido un evento.

## Simulación Orientada a Eventos Discretos

- Examina **solo los instantes** en que ocurren los eventos.
- El reloj "salta" de un evento al siguiente.
- Es más eficiente, ya que ignora los períodos de inactividad.

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos**
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Simulación: Métodos Numéricos vs. Analíticos

## Métodos Analíticos

- Emplean el razonamiento deductivo de las matemáticas para '**resolver**' el modelo.
- Es un enfoque puramente teórico.
- *Ejemplo:* Usar cálculo diferencial para encontrar el costo mínimo en un modelo de inventario.

## Métodos Numéricos (Simulación)

- Emplean procedimientos computacionales para analizar el modelo.
- Los modelos se '**ejecutan**' en lugar de resolverse.
- Se genera una historia artificial del sistema para estimar su rendimiento.

## Ejemplo: Cliente en un Banco

### Examples

Un ejercicio común para aprender a construir simulaciones de eventos discretos es modelar una cola simple.

Para el sistema de un cliente que llega a un banco para ser atendido por un cajero, podemos identificar:

- **Entidades:** La cola de clientes y los cajeros.



# Ejemplo: Cliente en un Banco

## Examples

Un ejercicio común para aprender a construir simulaciones de eventos discretos es modelar una cola simple.

Para el sistema de un cliente que llega a un banco para ser atendido por un cajero, podemos identificar:

- **Entidades:** La cola de clientes y los cajeros.
- **Eventos:** La llegada del cliente y la partida del cliente.

# Ejemplo: Cliente en un Banco

## Examples

Un ejercicio común para aprender a construir simulaciones de eventos discretos es modelar una cola simple.

Para el sistema de un cliente que llega a un banco para ser atendido por un cajero, podemos identificar:

- **Entidades:** La cola de clientes y los cajeros.
- **Eventos:** La llegada del cliente y la partida del cliente.
- **Estados del Sistema:** El número de clientes en la cola (0, 1, 2, ...) y el estado del cajero (ocupado o inactivo).

## Ejemplo: Cliente en un Banco

### Examples

Un ejercicio común para aprender a construir simulaciones de eventos discretos es modelar una cola simple.

Para el sistema de un cliente que llega a un banco para ser atendido por un cajero, podemos identificar:

- **Entidades:** La cola de clientes y los cajeros.
- **Eventos:** La llegada del cliente y la partida del cliente.
- **Estados del Sistema:** El número de clientes en la cola (0, 1, 2, ...) y el estado del cajero (ocupado o inactivo).
- **Variables Aleatorias Clave:** El tiempo entre llegadas de los clientes y el tiempo de servicio del cajero.

# Agenda

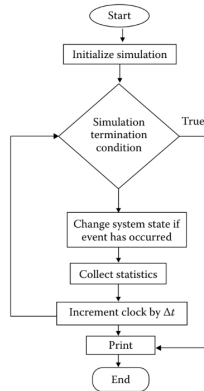
- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos**
  - **Ejemplo: Retiro en un Cajero Automático (ATM)**
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Componentes del Motor de Simulación DES

## El Motor de Simulación

Para gestionar la dinámica donde el tiempo 'salta' de un evento al siguiente, se requiere un motor de simulación con varios componentes fundamentales que trabajan en conjunto para procesar eventos en orden cronológico.

# Componentes del Motor de Simulación DES



# Elementos Esenciales del Motor DES I

## Reloj (Clock)

Lleva el registro del tiempo de simulación actual. En DES, el tiempo 'salta' de un evento al siguiente, avanzando hasta la hora de inicio del próximo evento en la lista.

## Lista de Eventos (Events list)

Mantiene una lista de eventos pendientes, ordenada cronológicamente. Es el corazón del motor, ya que dicta la secuencia de la simulación.

# Elementos Esenciales del Motor DES II

## Generadores de Números Aleatorios

Se utilizan para generar variables aleatorias (e.g., tiempos de servicio, tiempos entre llegadas) a partir de distribuciones de probabilidad. El uso de semillas permite que el comportamiento aleatorio sea repetible.

## Estadísticas (Statistics)

Realiza el seguimiento de las métricas de rendimiento del sistema que son de interés para el estudio (e.g., tiempo promedio en cola, utilización del servidor).



# Elementos Esenciales del Motor DES III

## Condición de Finalización (Ending condition)

Determina cuándo terminará la simulación. Puede ser un tiempo de simulación fijo, un número de eventos procesados o cuando una métrica alcanza un valor específico.

# Ejemplo: Simulación de un Retiro en un ATM

## ■ Estados del Sistema:

- $Q(t)$ : Número de clientes en la cola.
- $S(t)$ : Estado del servidor (0=inactivo, 1=ocupado).

## ■ Entidades: El cliente y el servidor.

## ■ Eventos:

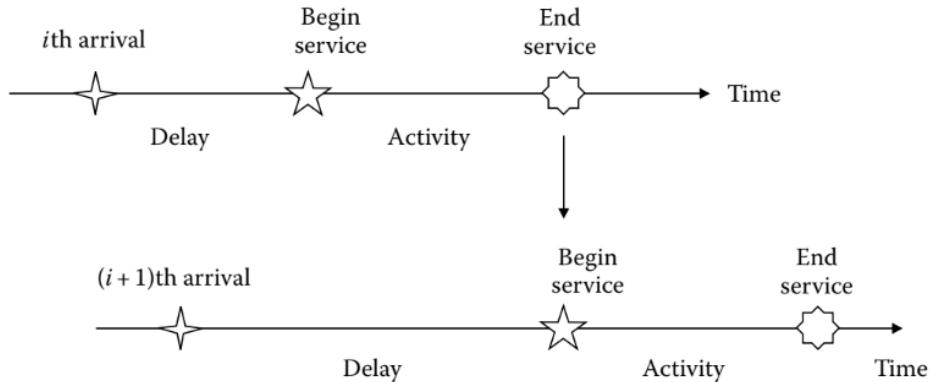
- **A**: Llegada (Arrival).
- **D**: Partida (Departure).
- **E**: Evento de detención de la simulación.

## ■ Actividades: El tiempo entre llegadas y el tiempo de servicio.

## ■ Retraso (Delay): El tiempo que un cliente pasa esperando en la cola.

Ejemplo: Retiro en un Cajero Automático (ATM)

## Ciclo de Vida de un Cliente en un ATM



# Lógica de los Eventos

---

**Algorithm 1** Lógica del Evento de Llegada

---

- 1: **Evento de Llegada** ocurre en Reloj =  $t$
  - 2: **if**  $S(t) == 1$  **then**
    - ▷ El servidor está ocupado
  - 3:    $Q \leftarrow Q + 1$ 
    - ▷ El cliente se une a la cola
  - 4: **else**
  - 5:    $S(t) \leftarrow 1$ 
    - ▷ El servidor se ocupa
  - 6:   Generar tiempo de servicio  $T_s$
  - 7:   Programar nueva partida en  $t + T_s$
  - 8: **end if**
  - 9: Generar tiempo entre llegadas  $T_a$
  - 10: Programar próxima llegada en  $t + T_a$
  - 11: Recopilar estadísticas
  - 12: *Llamar a la rutina de avance del tiempo*
-

# Lógica de los Eventos

---

**Algorithm 2** Lógica del Evento de Partida

---

- 1: **Evento de Partida** ocurre en  $\text{Reloj} = t$
  - 2: **if**  $Q(t) > 0$  **then** ▷ Hay clientes esperando
  - 3:      $Q \leftarrow Q - 1$  ▷ Un cliente sale de la cola para ser atendido
  - 4:     Generar tiempo de servicio  $T_s$
  - 5:     Programar nueva partida en  $t + T_s$
  - 6: **else**
  - 7:      $S(t) \leftarrow 0$  ▷ El servidor queda inactivo
  - 8: **end if**
  - 9: Recopilar estadísticas
  - 10: *Llamar a la rutina de avance del tiempo*
-

Ejemplo: Retiro en un Cajero Automático (ATM)

## Datos de Entrada y Resultados

### Tabla de Entrada

Table 2: Tiempo entre llegadas y tiempo de servicio de los clientes.

<b>Tiempo entre llegadas</b>	0	5	3	1	1
<b>Tiempo de servicio</b>	3	2	1	2	1

# Datos de Entrada y Resultados

## Tabla de Resultados (Traza)

Table 3: Resultados de la simulación para un mostrador de cajero automático.

Descripción	Reloj	Q(t)	S(t)	BMQ
1ra Llegada ( $T_a=0$ , $T_s=3$ )	0	0	1	0
1ra Partida	3	0	0	3
2da Llegada ( $T_a=5$ , $T_s=2$ )	5	0	1	3
2da Partida	7	0	0	5
3ra Llegada ( $T_a=8$ , $T_s=1$ )	8	0	1	5
3ra Partida	9	0	1	6
4ta Llegada ( $T_a=9$ , $T_s=2$ )	9	0	1	6
5ta Llegada ( $T_a=10$ , $T_s=1$ )	10	1	1	8
4ta Partida	11	0	1	8
5ta Partida	12	0	0	9

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones**
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización



# La Base de un Modelo Creíble

## Un Compromiso de Tiempo y Recursos

La recopilación y el análisis de datos de entrada es una de las tareas principales y más críticas en un proyecto de simulación.

## Principio GIGO (Garbage In, Garbage Out)

Datos de entrada incorrectos o defectuosos conducen a resultados de simulación erróneos y pueden llevar a tomar decisiones equivocadas.

# Procedimiento para el Modelado de Datos de Entrada

- 1 **Recolección de datos:** Recopilar datos del sistema real. Si no es posible, se puede recurrir al conocimiento de expertos para generar datos representativos.

# Procedimiento para el Modelado de Datos de Entrada

- 1 **Recolección de datos:** Recopilar datos del sistema real. Si no es posible, se puede recurrir al conocimiento de expertos para generar datos representativos.
- 2 **Identificar la distribución de los datos:** Se crea un histograma o una distribución de frecuencias para visualizar la forma y tendencia de los datos.

## Procedimiento para el Modelado de Datos de Entrada

- 1 **Recolección de datos:** Recopilar datos del sistema real. Si no es posible, se puede recurrir al conocimiento de expertos para generar datos representativos.
- 2 **Identificar la distribución de los datos:** Se crea un histograma o una distribución de frecuencias para visualizar la forma y tendencia de los datos.
- 3 **Seleccionar la familia de distribución y sus parámetros:** Se elige una distribución teórica (e.g., Exponencial, Normal) que parezca ajustarse a la forma de los datos y se estiman sus parámetros (e.g., media, desviación estándar).

## Procedimiento para el Modelado de Datos de Entrada

- 1 **Recolección de datos:** Recopilar datos del sistema real. Si no es posible, se puede recurrir al conocimiento de expertos para generar datos representativos.
- 2 **Identificar la distribución de los datos:** Se crea un histograma o una distribución de frecuencias para visualizar la forma y tendencia de los datos.
- 3 **Seleccionar la familia de distribución y sus parámetros:** Se elige una distribución teórica (e.g., Exponencial, Normal) que parezca ajustarse a la forma de los datos y se estiman sus parámetros (e.g., media, desviación estándar).
- 4 **Verificar la bondad del ajuste:** Se utilizan pruebas estadísticas (e.g., Chi-cuadrado) para determinar objetivamente qué tan bien la distribución seleccionada representa los datos reales.

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada**
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Selección de Distribuciones de Probabilidad

## Un Paso Crítico

La selección de una distribución de probabilidad adecuada es un paso que determina la validez del modelo. Cada distribución tiene características únicas y es apropiada para modelar diferentes tipos de procesos aleatorios.

# Para variables que toman valores enteros

## Distribución Binomial

Modela el número de éxitos  $x$  en  $n$  ensayos independientes.

- **Uso:** Artículos defectuosos en un lote; clientes que compran.
- **PMF:**  $P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$

## Distribución de Poisson

Modela el número de eventos en un intervalo fijo de tiempo o espacio.

- **Uso:** Llegadas por hora; fallas por día.
- **PMF:**  $P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}$



# Para variables que toman valores enteros

## Distribución Binomial Negativa

Modela el número de ensayos  $x$  para observar  $k$  éxitos.

- **Uso:** Ítems a inspeccionar para encontrar  $k$  defectuosos.
- **PMF:**  
$$P(X = x) = \binom{x-1}{k-1} p^k (1-p)^{x-k}$$

## Distribución Discreta Uniforme

Modela una situación donde todos los  $n$  resultados son igualmente probables.

- **Uso:** Lanzamiento de un dado; selección al azar.
- **PMF:**  $P(X = x) = \frac{1}{n}$

# Para variables que toman cualquier valor en un rango

## Distribución Normal (Gaussiana)

La famosa "campana", simétrica alrededor de la media  $\mu$ .

- **Uso:** Tiempos de proceso, errores de medición, dimensiones.

- **PDF:**  $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

## Distribución Lognormal

Si  $\ln(X)$  se distribuye normal. Tiene sesgo a la derecha.

- **Uso:** Duración de tareas, tiempos de reparación.

- **PDF:**  $f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$

# Para variables que toman cualquier valor en un rango

## Distribución Exponencial

Modela el tiempo entre eventos de un proceso de Poisson. No tiene "memoria".

- **Uso:** Tiempo entre llegadas, vida de componentes sin desgaste.
- **PDF:**  $f(x) = \lambda e^{-\lambda x}$

## Distribución Gamma

Muy flexible, con sesgo a la derecha. La Exponencial es un caso especial.

- **Uso:** Tiempos de espera, plazos de entrega.
- **PDF:**  $f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}$

# Para variables que toman cualquier valor en un rango

## Distribución de Erlang

Modela el tiempo total para que ocurran  $k$  eventos exponenciales.

- **Uso:** Tiempo de un proceso con  $k$  fases secuenciales.
- **PDF:**  $f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}$

## Distribución de Weibull

Clave en análisis de fiabilidad. Modela tasas de fallo decrecientes, constantes o crecientes.

- **Uso:** Tiempo hasta el fallo de componentes.
- **PDF:**  $f(x) = \frac{\alpha}{\beta} \left(\frac{x}{\beta}\right)^{\alpha-1} e^{-(x/\beta)^\alpha}$

# Para variables que toman cualquier valor en un rango

## Distribución Beta

Versátil, definida en  $[0, 1]$ . Puede adoptar muchas formas.

- **Uso:** Modelar proporciones o porcentajes.
- **PDF:**  $f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$

## Distribución Triangular

Útil cuando solo se conocen el mínimo ( $a$ ), máximo ( $b$ ) y más probable ( $c$ ).

- **Uso:** Duración de actividades, opiniones de expertos.

# ¿Y si ninguna distribución teórica se ajusta?

## Distribución Empírica

No es una distribución teórica. Se construye directamente a partir de los datos históricos observados. Se puede representar como un histograma o una CDF escalonada.

- **Uso:** Se utiliza cuando ninguna de las distribuciones estándar proporciona un ajuste adecuado, pero se dispone de una cantidad suficiente de datos históricos.

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Generación de Números Aleatorios

## La Base de la Estocasticidad

Los números aleatorios son fundamentales para simular sistemas de eventos discretos. Permiten modelar la incertidumbre y la variabilidad inherentes a los sistemas del mundo real.



# Distribución Uniforme y el Método Congruencial Lineal

Una secuencia de números en  $[a, b]$  donde cada número tiene la misma probabilidad de ocurrir.

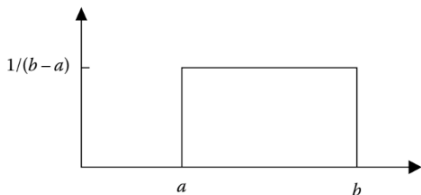


Figure 4: Densidad de probabilidad de una distribución uniforme.

## Método Congruencial Lineal

El método numérico más común para generar enteros aleatorios uniformes se basa en:

$$u_{k+1} = (\alpha u_k + \gamma) \pmod{\beta} \quad (1)$$

Donde  $\alpha$  (multiplicador),  $\gamma$  (incremento) y  $\beta$  (módulo) son enteros predefinidos.

## Generación de Números con Distribución Gaussiana

Es la distribución no uniforme más utilizada, caracterizada por su forma de campana centrada en la media  $\mu$ .

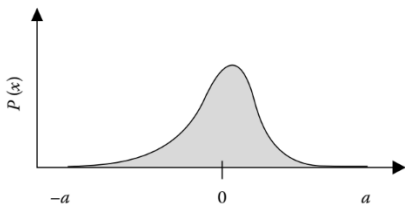


Figure 5: Densidad de probabilidad de una distribución Gaussiana.

### Método de Box-Muller

Permite transformar un par de números uniformes  $(u_1, u_2)$  en un par de números normales estándar  $(x_1, x_2)$ :

$$x_1 = \sqrt{-2 \ln u_1} \cos(2\pi u_2)$$

$$x_2 = \sqrt{-2 \ln u_1} \sin(2\pi u_2)$$

## Generación de Números Uniformes

## Función rand

Genera números aleatorios distribuidos uniformemente en el intervalo  $(0, 1)$ .

```
1 % Un solo numero entre 0 y 1
2 num = rand;
3
4 % Un vector fila de 1x5
5 vec = rand(1, 5);
6
7 % Para generar en un intervalo [a, b], ej: [10, 50]
8 a = 10;
9 b = 50;
10 num_escalado = a + (b-a) * rand;
```

# Generación de Números Normales

## Función randn

Genera números aleatorios de la distribución normal estándar ( $\mu = 0, \sigma = 1$ ).

```
1 % Una matriz de 3x3 de N(0,1)
2 matriz_normal = randn(3, 3);
3
4 % Para generar de N(mu, sigma), ej: N(15, 2.5)
5 media = 15;
6 desv_est = 2.5;
7 datos = media + desv_est * randn(1, 1000);
8 histogram(datos); % Visualizar
```

# Generación de Enteros Aleatorios

## Función randi

Genera números enteros distribuidos uniformemente en un rango.

```
1 % Un entero entre 1 y 10 (simula un dado)
2 dado = randi(10);
3
4 % Matriz 4x4 de enteros entre 50 y 100
5 rango = [50, 100];
6 matriz_enteros = randi(rango, 4, 4);
```

# Control del Generador Aleatorio

## Función rng

Permite controlar la semilla del generador para poder reproducir exactamente la misma secuencia de números aleatorios, lo cual es crucial para depurar y validar modelos.

```
1 % Establecer la semilla a un valor específico
2 rng(0);
3 secuencia1 = rand(1, 3)
4
5 % Restablecer la semilla al mismo valor
6 rng(0);
7 secuencia2 = rand(1, 3)
8 % secuencia2 sera identica a secuencia1
```

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

Continúa de Eventos Discretos Mediante Algoritmos de Eventos

# Introducción a SimEvents

## ¿Qué es SimEvents?

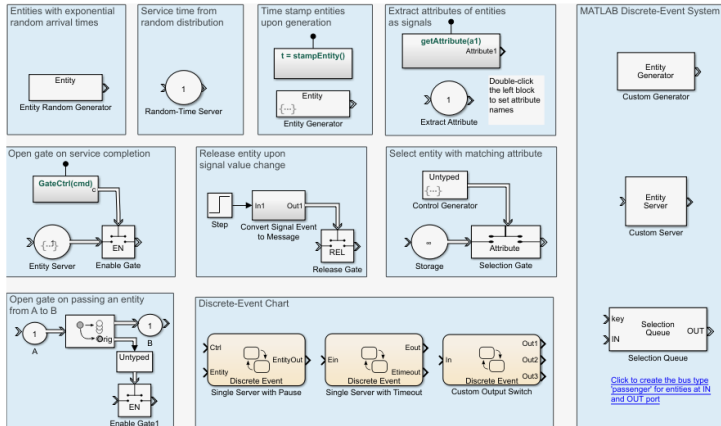
SimEvents es una extensión de Simulink que permite el modelado y la simulación de sistemas de eventos discretos. Es la herramienta ideal para diseñar y analizar sistemas cuya dinámica está gobernada por eventos asíncronos, como la llegada de clientes, la transmisión de paquetes de datos o el fallo de una máquina.

## Componentes Principales

La biblioteca de SimEvents proporciona un conjunto de bloques predefinidos que representan componentes comunes como generadores de entidades, colas, servidores y conmutadores.



# Plantillas para Escenarios Comunes



## Ejemplo: Creación de un Modelo D/D/1

### Escenario

Se creará un modelo de un sistema de colas simple donde camiones (entidades) llegan a una gasolinera para llenar sus tanques. Este sistema se conoce como **\*\*D/D/1\*\***, lo que indica una tasa de llegada determinista, una tasa de servicio determinista y un único servidor.

### Pasos Básicos de Construcción

- 1 Abrir un nuevo modelo en Simulink.
- 2 Abrir la biblioteca de SimEvents (comando `simevents`).
- 3 Arrastrar los bloques necesarios al modelo.

## Modelo D/D/1: Bloques y Conexiones

### Bloques Utilizados

- **Entity Generator:** Modela la llegada de los tanques.
- **Entity Queue:** Modela la espera de los tanques.
- **Entity Server:** Modela el proceso de llenado.
- **Entity Terminator:** Modela la partida de los tanques.
- **Scope:** Visualiza las señales de

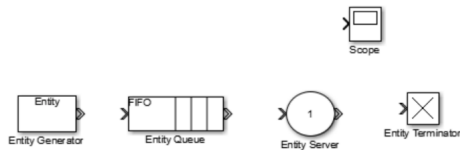
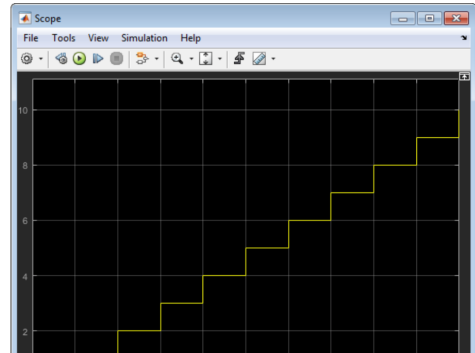


Figure 7: Bloques básicos para el modelo de colas.

# Modelo D/D/1: Configuración y Resultados

## Configuración Clave

- En **Entity Generator**, el **Period** se deja en 1 (un tanque llega cada segundo).
- En **Entity Server**, el **Service time** se deja en 1.0 (el servidor tarda un segundo en llenar el tanque).
- Se conecta la salida de estadísticas 'd' (entidades que han partido) del servidor al Scope.



# Visualización de Estadísticas

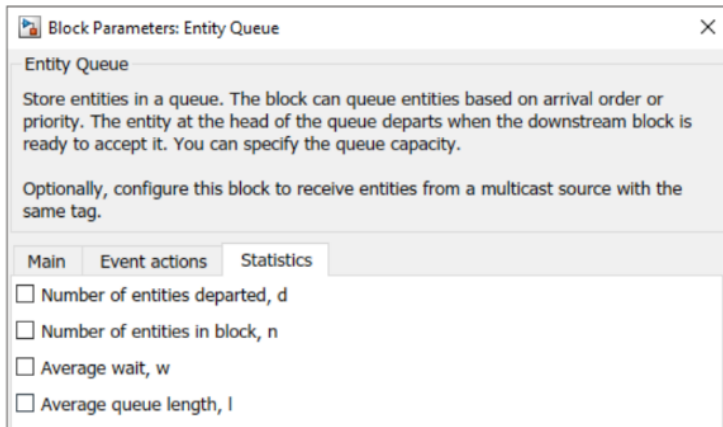
## ¿Por qué visualizar estadísticas?

El propósito principal de una simulación es comprender el sistema para informar decisiones. Los datos estadísticos (tasas de producción, tiempos de espera, utilización) son clave para interpretar el comportamiento del modelo.

## ¿Cómo acceder a las estadísticas?

Muchos bloques de SimEvents tienen una pestaña de **Estadísticas**. Al seleccionar una casilla, se crea un nuevo puerto de salida en el bloque para esa señal estadística.

# Visualización de Estadísticas



# Análisis de Escenarios: Tasa de Llegada vs. Tasa de Servicio

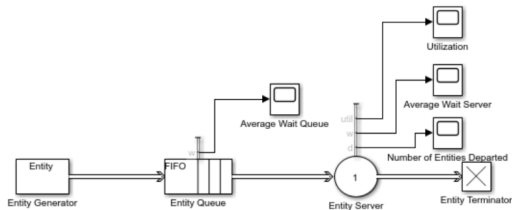


Figure 11: Modelo modificado para visualizar estadísticas de espera y utilización.

Si Llegadas  $\geq$  Servicio

(Ej: Periodo de Llegada = 0.3s)

Si Llegadas  $\leq$  Servicio

(Ej: Periodo de Llegada = 1.1s)

# Acciones de Eventos

## ¿Qué son las Acciones de Eventos?

SimEvents permite crear acciones personalizadas cuando ocurre un evento. Estas acciones, escritas en código de MATLAB, permiten cambiar los atributos de las entidades de forma dinámica.

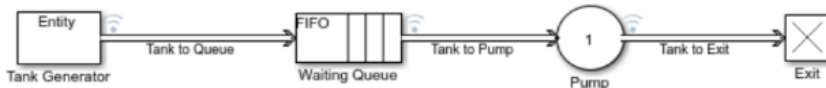


# Acciones de Eventos

## Ejemplo: Llenado de Tanques

Se modela una gasolinera donde los tanques llegan con un nivel de gasolina aleatorio y se llenan con una cantidad fija.

- **Acción "Generate":** Se asigna un nivel de gasolina inicial aleatorio a cada tanque que llega.
- **Acción "Service complete":** Se incrementa el nivel de gasolina del tanque al completar el servicio.



# Integración con Simulink Functions

## ¿Qué son las Simulink Functions?

Se pueden usar bloques de **Simulink Function** para obtener atributos de las entidades de SimEvents, pasarlos a componentes de Simulink para su procesamiento y luego devolverlos al modelo de eventos discretos.

# Integración con Simulink Functions

## Aplicaciones Típicas

- **Timestamping:** Registrar el tiempo de generación y de servicio para calcular la duración total de una entidad en el sistema.
- **Paso de Atributos:** Enviar el valor de un atributo a un subsistema de Simulink para realizar cálculos complejos.
- **Importar Datos:** Usar datos de una hoja de cálculo para definir parámetros del modelo, como los tiempos entre llegadas.
- **Notificación de Eventos:** Crear un evento para notificar a un bloque de enrutamiento (switch) cuando se complete un proceso].

# Agenda

- 1 Introducción
- 2 Definiciones
- 3 Sistemas de Colas (*Queuing Systems*)
- 4 Simulación de Sistemas de Eventos Discretos
- 5 Componentes de la simulación de sistemas de eventos discretos
  - Ejemplo: Retiro en un Cajero Automático (ATM)
- 6 Modelado de Datos de Entrada y Distribuciones
- 7 Familias de Distribuciones para Datos de Entrada
- 8 Generación de Números Aleatorios
- 9 Simulación de eventos discretos con SimEvents en MATLAB-Simulink
  - Creación de un Modelo simple de Eventos Discretos
  - Exploración de Estadísticas y Visualización

# Integración de SimEvents con Simulink

## Combinando lo Mejor de Dos Mundos

Se pueden usar bloques de **Simulink Function** para crear una poderosa sinergia entre el mundo de los eventos discretos y el mundo del tiempo continuo.

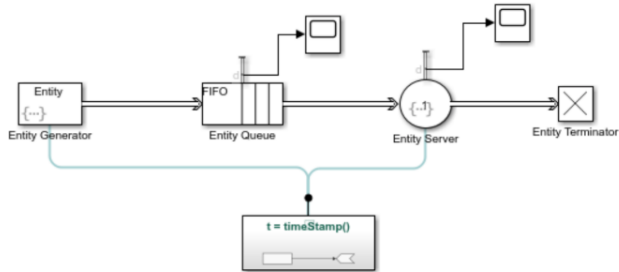
## Casos de Uso Principales

- Estampar el tiempo de las entidades (*Timestamping*).
- Pasar atributos de entidad a componentes de Simulink para procesamiento.
- Crear eventos de notificación para controlar el flujo (enrutamiento).
- Importar datos de fuentes externas como hojas de cálculo.

## Estampado de Tiempo de Entidades (Timestamping)

# Ejemplo: Timestamping

Este modelo utiliza una **Simulink Function** para registrar el tiempo en el que una entidad es generada y el tiempo en que completa su servicio, permitiendo calcular la duración total que pasó en el sistema.



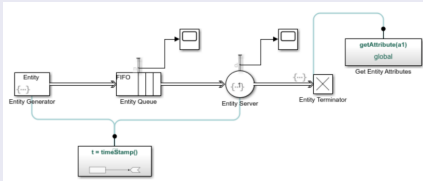
## Ejemplo: Timestamping

- Acción 'Generate' del Generador: Llama a la función para registrar el `TimeStampGeneration`.
- Acción 'Service complete' del Servidor: Llama a la función para registrar el `TimeStampServiceComplete` y calcula el `TotalTime`.

# Otros Patrones de Integración con Simulink

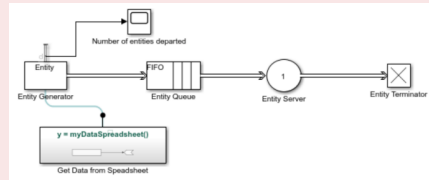
## Pasar Atributos

Se puede pasar el valor de un atributo a un componente de Simulink (como un bloque Gain) para realizar cálculos.



## Importar Datos

Se pueden usar datos de una hoja de cálculo para definir parámetros, como los tiempos entre llegadas de entidades.

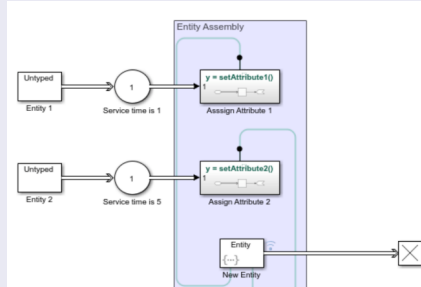




# Patrones Avanzados de Integración

## Ensamblaje de Entidades

Es posible tomar atributos de diferentes tipos de entidades y combinarlos para crear una nueva entidad ensamblada.



# Patrones Avanzados de Integración

## Enrutamiento por Notificación

Una Simulink Function puede generar un evento que notifica a un bloque de enrutamiento (*switch*) para dirigir las entidades según una lógica de control de calidad.

