

Conceptos clave de Simulación de Eventos Discretos

Juliana Quintana-Rojas

Departamento de Ingeniería Industrial
Universidad de Los Andes

Simulación de eventos discretos

¿Cómo evoluciona el sistema?

- El sistema **evoluciona en el tiempo**
- Las **variables de estado** describen el estado del sistema en cualquier momento
- Ocurren **eventos** de acuerdo con algún patrón (probabilístico)
- Cada **evento modifica** el estado del sistema
- El **reloj de simulación** avanza en pasos discretos, solo cuando ocurre un evento: es innecesario revisarlo si no hay cambios de estado

Ejemplo: sistema de inventarios

- Variable de estado: **nivel de inventario**
- Se realizan solicitudes a un CEDI, programadas para llegar después de un **tiempo incierto**
- Al llegar la solicitud, **aumenta** el nivel de inventario
- Las **ventas reducen** el inventario
- Los tiempos entre ventas son **incierto**

Eventos

Un evento es una ocurrencia instantánea que puede cambiar el estado del sistema

Exógenos

Sucesos que ocurren fuera del sistema pero que lo afectan

Ej. Llegada de un auto al drive-thru

Endógenos

Actividades y sucesos que ocurren dentro del sistema

Ej. finalización de la orden en el tablero de menú

Primarios

Salida programada

● Directamente en la lista de eventos futuros

RELACIÓN CON LAS ACTIVIDADES

Una actividad tiene una duración conocida al inicio (espera incondicional); el final de dicha actividad genera **un evento primario**

Condicionales

Solo ocurren si se cumplen ciertas condiciones lógicas

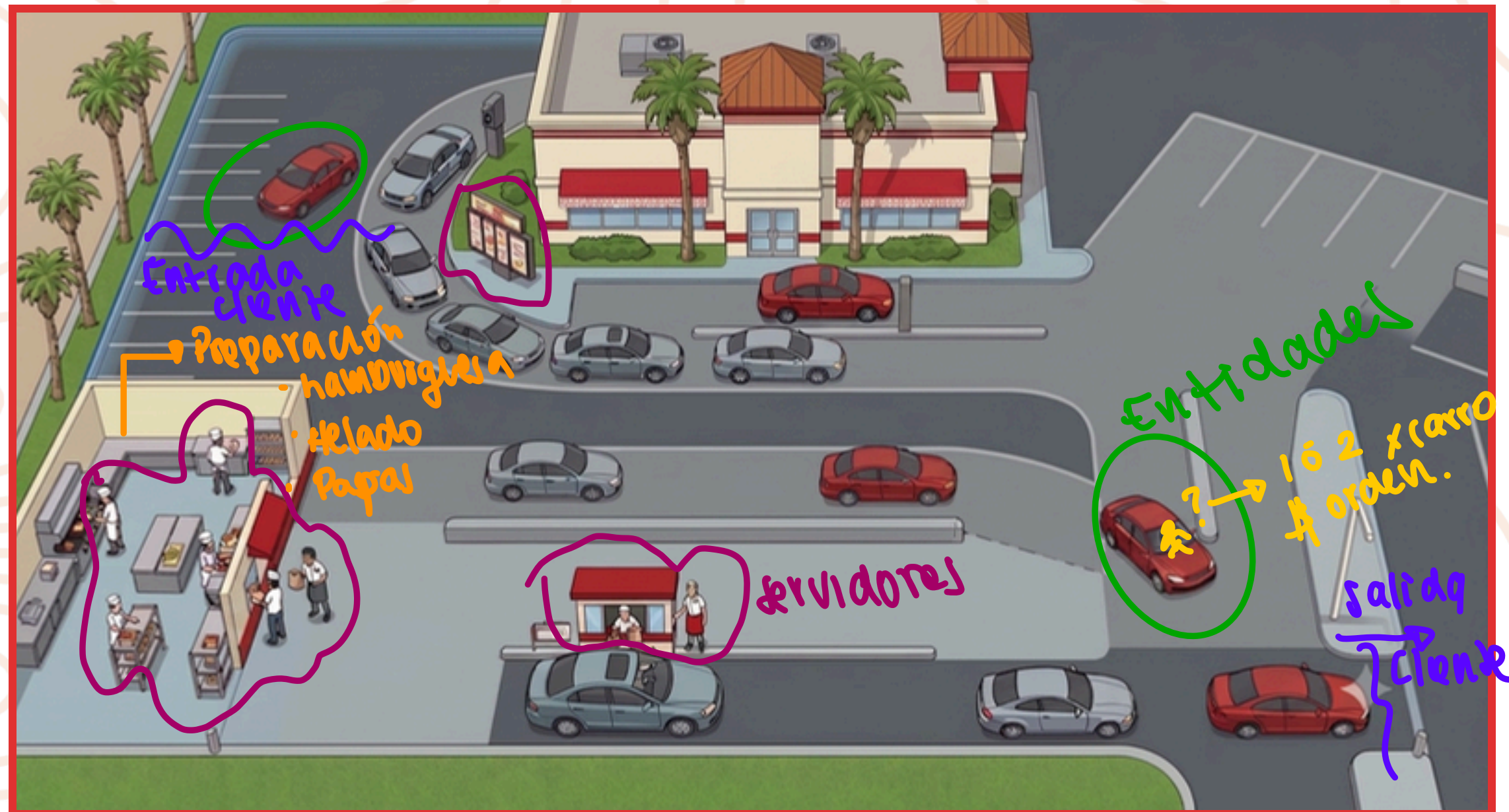
Ej. iniciar una orden cuando el tablero está libre y hay un auto en la fila

Drive - thru

El sistema representa un restaurante de comida rápida con servicio drive-through. Los vehículos deciden si ingresan según la longitud de la fila; después realizan el pedido, avanzan cuando existe espacio y esperan hasta que el automóvil y su orden coincidan en la ventanilla de entrega. Las salidas del sistema son clientes satisfechos y clientes que abandonan el restaurante por una fila demasiado larga.



Componentes del sistema



ENTIDADES

Autos: objetos de interés que causan cambios en el estado del sistema

ATRIBUTOS

Propiedades únicas de cada entidad. Ej. valor de su pedido

RECURSOS

Prestan un servicio a las entidades: tablero de menú, cocina, ventanilla de pago y entrega

ACTIVIDADES

Procesos con duración conocida al inicio: tiempo de orden, de preparación, de cobro y entrega

EVENTOS

Ocurrencias instantáneas que cambian el estado: llegada y salida del cliente

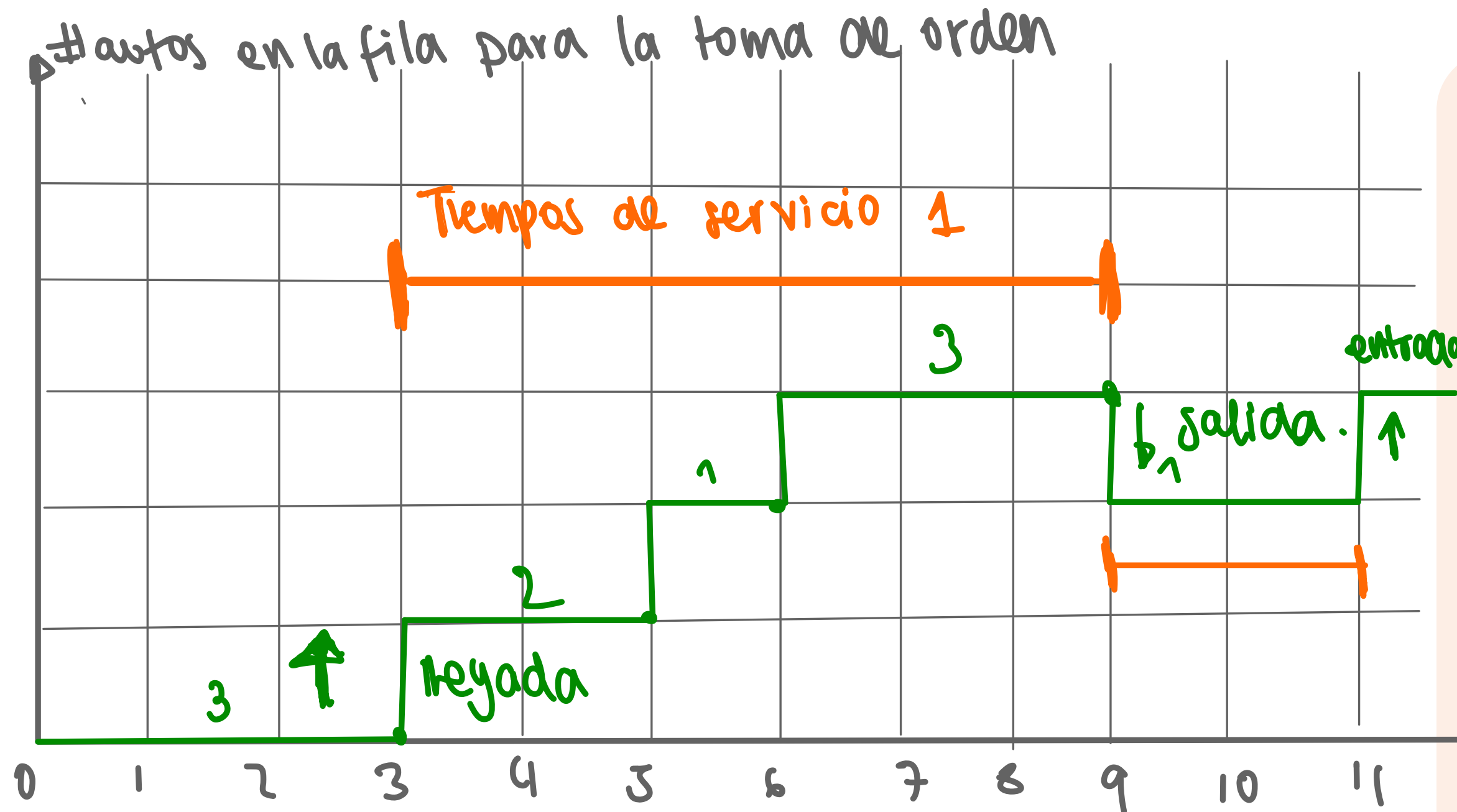
ESTADO DEL SISTEMA

Variables para describir el sistema: tiempo actual y número de autos en la fila

Elementos de la SED en el drive-thru

Elemento	Definición	En el drive-thru
Sistema	Conjunto de entidades que interactúan en el tiempo para alcanzar un objetivo	In-n-Out Restaurante con servicio drive-thru
Modelo	Representación abstracta que captura las relaciones del sistema	Llegada autos, toma pedido, orden preparación, pago y entrega
Variables de estado	Describen el estado del sistema en cualquier momento del tiempo	# autos en fila, toma pedido disponible • no. # autos espera entrega.
Entidad	Objeto del sistema que requiere describirse explícitamente	Autos (clientes)
Atributos	Propiedades de una entidad	Valor pedido Cant. ítems orden
Evento	Suceso que cambia el estado del sistema	Llegado, salida auto fin de preparación orden
Noticia de evento	Registro de ocurrencia de un evento	{ tipo evento; tiempo ocurrencia }
Lista de eventos futuros (FEL)	Noticias de eventos futuros, ordenadas por momento de ocurrencia	Próximas llegadas, fin órdenes programadas
Actividad	Duración de tiempo definida, conocida al iniciar (determinística o aleatoria)	Tiempo preparación de un producto
Retraso (delay)	Duración no definida; depende de la ocurrencia de otros eventos	Espera en la fila Tiempo total en el sistema

Ejemplo manual: el drive-thru



Tiempos entre llegadas: $T_1 = 3, T_2 = 2, T_3 = 1, T_4 = 3, T_5 = 1, T_6 = 2$

Tiempos de orden en el tablero: $S_1 = 6, S_2 = 3$

Supuestos del ejercicio

- Un solo punto de orden: el **tablero de menú**
- $X(t)$: autos en la fila del tablero (en orden + en espera)
- Eventos: **llegada** de un auto y **fin de la orden**
- El primer auto llega en $t = 3$
- Si el tablero está ocupado, el auto **espera en la fila**
- $X(t)$ sube 1 con cada llegada y baja 1 con cada fin de orden

Rastreo del drive-thru con la lista de eventos

t	Lista de eventos	Next
–	{	0
0	{ <u>llegada: 3</u> }	3 ✓
3	{ <u>llegada: 3+2 = 5</u> ; fin de orden: 3+6 = 9}	5 ✓
5	{ <u>llegada: 5+1 = 6</u> ; fin de orden: 9}	6 ✓
6	{llegada: 6+5 = 11; <u>fin de orden: 9</u> }	9 ✓
9	{ <u>llegada: 11</u> ; fin de orden: <u>9+3 = 12</u> }	11 ✓
11	...	...

reloj salta al tiempo de evento + 'inminente (0, 3, 5, 6, 9, 11), no se revisan los instantes en lo que nada ocurre

En $t=9$ termina la orden auto 1 e inicia la orden auto 2 | Fin orden A2
 $9 + S_2 = 12$

Reloj de simulación

Es la variable que define el **valor actual del tiempo simulado**



Naturaleza

No existe una relación directa entre el tiempo que tarda el computador en ejecutar el modelo y el tiempo que avanza el reloj de simulación



Funcionamiento

El reloj no avanza de forma continua: **salta de un tiempo de evento al siguiente**, ignorando los periodos de inactividad

Lista de eventos futuros (FEL)

El «calendario» de la simulación: garantiza que los eventos ocurran en el **orden cronológico correcto**.

- **Definición:** lista de avisos de eventos (event notices), ordenados por su tiempo de ocurrencia
- **Aviso de evento:** registro que contiene, como mínimo, el **tipo de evento** y el **tiempo** en que debe ocurrir
- **Estructura de datos:** en simulaciones complejas se implementa como una **lista ligada** (simple o doble), que permite insertar y eliminar registros eficientemente
- **Gestión:** el sistema siempre toma de la parte superior el evento con el tiempo más bajo: el **evento inminente**

Evento inminente (sale primero)

llegada
 $t = 5$

fin de orden
 $t = 9$

llegada
 $t = 11$

fin de orden
 $t = 12$

Cada aviso registra el tipo de evento y su tiempo de ocurrencia, ordenados de menor a mayor

Avance al siguiente evento (NETA)

El reloj salta directamente al tiempo del evento más inminente de la FEL. Es el esquema estándar de la industria.

- 1 Eliminar** de la FEL el aviso del evento inminente
- 2 Avanzar el reloj** al tiempo de dicho evento
- 3 Ejecutar el evento:** actualizar el estado del sistema, los atributos de las entidades y las colas
- 4 Programar eventos futuros (bootstrapping):** generar nuevos tiempos de actividades e insertar sus avisos en la FEL en la posición cronológica correcta
- 5 Actualizar estadísticas:** recolectar datos para los contadores de desempeño

EJEMPLO DE INTEGRACIÓN — DRIVE-THRU

La llegada de un auto es un **evento exógeno** que se programa en la FEL usando un generador de números aleatorios. Al procesarse esa llegada, NETA avanza el reloj al tiempo de llegada, actualiza el estado («un auto más en la fila») y programa el siguiente evento («fin de la orden en el tablero de menú») sumando la duración de la actividad al tiempo actual del reloj.

Dos formas de avanzar el reloj

ESTÁNDAR

NETA

Avance al siguiente evento

- El reloj **salta** al tiempo del evento más inminente de la FEL
- Solo se procesan los instantes en los que **cambia el estado** del sistema
- No desperdicia cómputo en periodos de inactividad
- Sin pérdida de precisión: los eventos ocurren en su **tiempo exacto**

POCO COMÚN EN SED

FITA

Avance por incrementos fijos

- El reloj avanza en pasos de tamaño constante (Δt)
- Tras cada incremento se verifica si ocurrió algún evento; los eventos se procesan **como si hubieran ocurrido al final del intervalo**
- Introduce **errores de precisión** en los tiempos de ocurrencia
- Ineficiente: revisa intervalos donde **no pasa nada**

Vistas del mundo (world views) en SED

Marcos conceptuales que definen la orientación para organizar la lógica del modelo de simulación

1

Programación de eventos

El analista se concentra en los **eventos** y su efecto sobre el estado del sistema; la simulación ejecuta eventos en orden cronológico con la FEL. Método estándar en lenguajes de propósito general como C.

2

Interacción de procesos

Describe el **ciclo de vida de una entidad** al fluir por el sistema: procesos (eventos, actividades y retrasos) donde las entidades demandan recursos y hacen cola si están ocupados. El enfoque más popular en el software de simulación moderno.

3

Revisión de actividades

Se centra en las **actividades** y las condiciones lógicas que permiten que inicien; en cada avance del reloj se revisan todas las condiciones. La variante de **tres fases** combina el avance por eventos con el escaneo de actividades condicionales.

Visión de programación de eventos

- Las **variables de estado** describen el sistema
- Los **eventos** afectan las variables de estado
- Los eventos futuros se programan explícitamente en la **lista de eventos futuros**

En el drive-thru:

Estado: autos en la fila y tablero libre u ocupado. Eventos: llegadas y fines de orden, programados en la FEL.

t	Lista de eventos	Next
–	{	0
0	{llegada: 3}	3
3	{llegada: 5; fin orden: 9}	5
5	{llegada: 6; fin orden: 9}	6
6	{llegada: 11; fin orden: 9}	9
9	{llegada: 11; fin orden: 12}	11
11	...	12

Visión de interacción de procesos

Describe la **experiencia del auto** en su paso por el drive-thru: un proceso es la lista de eventos, actividades y retrasos que ocurren secuencialmente y definen su ciclo de vida

#	Paso del proceso	Tipo	Interacción
1	Llegada del auto	Evento	—
2	Espera en la fila	Retraso	—
3	Inicio de la orden	Evento	—
4	Orden en el tablero de menú	Actividad	—
5	Fin de la orden	Evento	Inicia orden siguiente auto en la fila

Es el enfoque **más popular en el software de simulación moderno** por resultar más intuitivo: el modelo se define desde la perspectiva de la entidad que fluye por el sistema

Visión de revisión de actividades

Actividades tipo B (no condicionales)

Actividades o eventos que ocurren **sin condiciones**.

En el drive-thru: las llegadas de los autos y el fin de cada orden.

Actividades tipo C (condicionales)

Retrasos: su ocurrencia **depende de otros eventos o condiciones**.

En el drive-thru: la espera en la fila, que termina cuando el tablero queda libre.

Enfoque de tres fases

Fase 1: tomar el siguiente evento de la lista y avanzar el reloj; tomar otros eventos que ocurran al mismo tiempo.

Fase 2: ejecutar todas las actividades tipo B retiradas de la lista de eventos.

Fase 3: revisar las condiciones que pueden activar actividades tipo C y ejecutar las que las cumplan.

Ejemplo: en $t = 9$ termina la orden del auto 1 (fases 1 y 2); en la fase 3, la espera del auto 2 cumple su condición —tablero libre— e inicia su orden, cuyo fin se programa en $t = 12$