

**Capítulo 2**  
*Como funcionam a modelagem  
e a simulação discreta de  
sistemas*  
**Aula 3**

# Métodos de Modelagem e Visão da Realidade

---

- ◆ **Visão da realidade:** forma de ver (entender) os sistemas, existentes ou não, visando sua modelagem. Existem, basicamente, três diferentes “visões” e/ou métodos de modelagem.

1. Modelagem por **eventos**;
2. Modelagem por **atividades**;
3. Modelagem por **processos**.

# Modelagem por Eventos

---

- ◆ O sistema é modelado pela identificação de seus **eventos característicos** (os quais são incondicionais), dependendo unicamente do tempo de simulação.
- ◆ **Rotinas descrevem as mudanças de estado** que podem ocorrer no sistema em pontos discretos no tempo, de acordo com a ocorrência dos eventos. Estas rotinas descrevem ações relacionados a ocorrência dos eventos.
- ◆ O processo de simulação evolui ao longo do tempo pela **execução dos eventos selecionados de uma pilha ou calendário de eventos**, cronologicamente ordenados. É executado o evento do topo da pilha (tempo mais próximo do tempo corrente de simulação).

# Modelagem por Atividades

---

- ◆ A estratégia de busca do *próximo evento da lista* é baseada tanto no **tempo programado** de ocorrência como em **testes condicionais**.
  - ✓ aplicável a sistemas cujas mudanças de estado dependem da ocorrência de eventos condicionados, isto é, outras condições, além do tempo devem ser verdadeiras.
  - ✓ a busca sobre o próximo evento deve considerar ambas as situações: **tempo de simulação** e **quaisquer outras condições favoráveis ao *disparo*** de um evento.
  - ✓ A monitoração de situações de busca de condições de início de eventos é típica da modelagem por atividades.

# Modelagem por Processos

---

- ◆ Os sistemas são vistos principalmente do **ponto de vista das entidades**.
- ◆ O programa **monitora cada entidade**, individualmente.
- ◆ Sempre haverá uma **entidade controlando a sequência de eventos**.
- ◆ O **controle é transferido de uma entidade para outra**, na medida que eventos futuros possam ser programados para cada uma delas.
- ◆ Estes procedimentos se repetem com **todas as entidades que estejam no sistema até que estas o deixem**, quando realizarem todas as suas atividades.

# Mecanismos de Avanço do Tempo

---

- ◆ A natureza dinâmica dos modelos de simulação implica em manter um **constante acompanhamento do valor do tempo simulado**, enquanto a simulação avança.
- ◆ É necessário também que o programa de simulação possua um **mecanismo para avançar o tempo simulado** de um valor para outro.
- ◆ A variável que guarda o tempo atualizado de simulação é chamada de *relógio da simulação*.

# Mecanismos de Avanço do Tempo

---

- ◆ Dois principais mecanismos de avanço do tempo aparecem nas diversas linguagens de simulação:
  - ✓ *avanço do tempo com incremento fixo*
  - ✓ *avanço do tempo para o próximo evento*
- ◆ O último é amplamente utilizado tanto por programas comerciais de simulação como por aqueles montados sobre uma linguagem de programação de propósito geral como C, Delphi, VB, Java, etc.
- ◆ Nossa referencia será ao **avanço do tempo relacionado ao próximo evento.**

# Método da Programação de Eventos

---

- ◆ O método da **programação de eventos** considera a execução de um código que contém um **laço sobre uma lista de eventos**.
- ◆ Pode-se empregar estruturas de dados, tais como **listas ou pilhas encadeadas**.
- ◆ Um dos principais elementos dos algoritmos voltados ao método da programação de eventos é a ***lista de eventos futuros (LEF)***.



# Calendário ou Lista de Eventos Futuros (LEF)

- ◆ O **calendário de eventos**, é uma **lista** contendo todos os eventos programados para ocorrerem no futuro, isto é, em algum momento posterior ao tempo atual do relógio da simulação.

RELÓGIO	Estado sistema	Entidades e atributos	Conj. 1	Conj. 2	...	Calendário de Eventos Futuros (LEF)	Estatísticas e Contadores
$t$	( $x, y, z, \dots$ )					( $E_3, t_1$ ) - Evento tipo 3 no tempo $t_1$ ( $E_1, t_2$ ) - Evento tipo 1 no tempo $t_2$ . .	

# Algoritmo de Avanço do Tempo

---

- ◆ A seqüência de ações que um simulador precisa realizar para avançar o relógio da simulação e criar uma nova *imagem* do sistema
- ◆ Baseia-se no *algoritmo para avanço do tempo com base na programação de eventos*, o qual faz uso do calendário ou Lista de Eventos Futuros (LEF).

# Algoritmo de Avanço do Tempo Baseado na Programação de Eventos

**Imagem do sistema no tempo  $t$**

RELÓGIO	Estado do Sistema	...	Lista de Eventos Futuros - (LEF)	...
$t$	$(0, 0, 0)$		$(E_3, t_1)$ Evento tipo 3 a ocorrer no tempo $t_1$ $(E_2, t_2)$ Evento tipo 2 a ocorrer no tempo $t_2$ $(E_1, t_3)$ Evento tipo 1 a ocorrer no tempo $t_3$     $(E_2, t_n)$ Evento tipo 2 a ocorrer no tempo $t_n$	

Algoritmo para avanço do tempo com base na programação de eventos

1. Remova o evento iminente da LEF (evento 3, tempo  $t_1$ );
2. Avance o RELÓGIO para o tempo do evento iminente (de tempo  $t$  para  $t_1$ );
3. Execute o evento iminente atualizando (na medida do necessário): o estado do sistema, os atributos das entidades e os membros de conjuntos;
4. Gere futuros eventos (se necessário) e coloque-os na LEF na posição correta. (Exemplo: Evento 1 que ocorrerá no tempo  $t^*$ , onde  $t_2 < t^* < t_3$ );
5. Atualize estatísticas acumuladas e contadores.

**Imagem do sistema no tempo  $t_1$**

RELÓGIO	Estado do Sistema	...	Lista de Eventos Futuros - (LEF)	...
$t_1$	$(0, 1, 0)$		$(E_2, t_2)$ Evento tipo 1 a ocorrer no tempo $t_2$ $(E_1, t^*)$ Evento tipo 1 a ocorrer no tempo $t^*$ $(E_1, t_3)$ Evento tipo 1 a ocorrer no tempo $t_3$     $(E_2, t_n)$ Evento tipo 2 a ocorrer no tempo $t_n$	

# Funcionamento de um Programa de Simulação (orientado a eventos)

---

- ◆ Na simulação discreta as **mudanças de estado** acontecem em pontos discretos no tempo (**eventos**).
- ◆ A cada mudança (evento) uma *imagem representa o estado do sistema naquele instante*.
- ◆ A sequência cronológica das imagens representa a **evolução do sistema no tempo**.
- ◆ Juntando-se todas as imagens produzidas, temos uma espécie de “**filme**” ou “**histórico**” dos fatos ou eventos ocorridos, assim como suas implicações sobre o sistema.

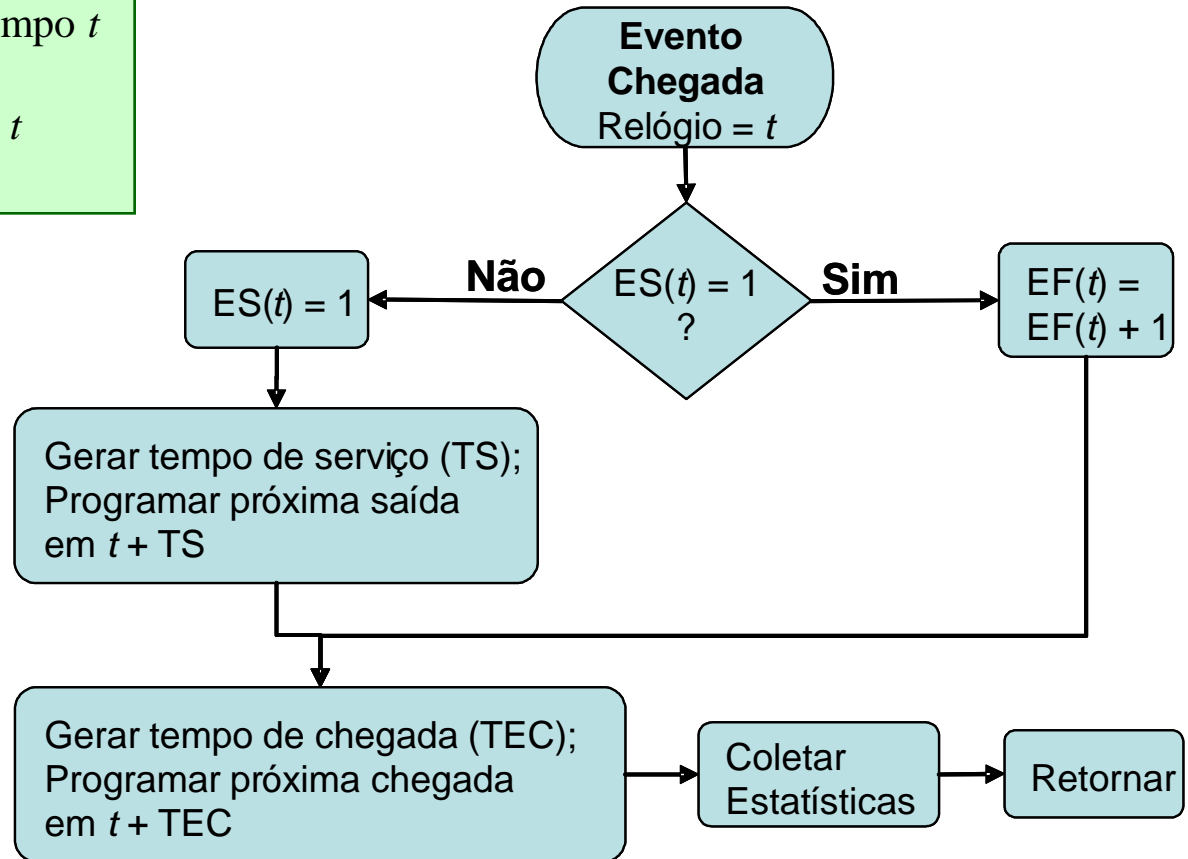
# Funcionamento de um Programa de Simulação (orientado a eventos)

---

- ◆ Principais elementos de um programa computacional voltado a simulação de modelos de mudança discreta.
  - ✓ *Rotina Principal;*
  - ✓ *Rotina de Inicialização;*
  - ✓ *Rotinas de Eventos (Ex. Evento Chegada e Saída);*
  - ✓ *Rotina de Avanço do Tempo;*
  - ✓ *Biblioteca de Funções;*
  - ✓ *Gerador de Relatórios.*

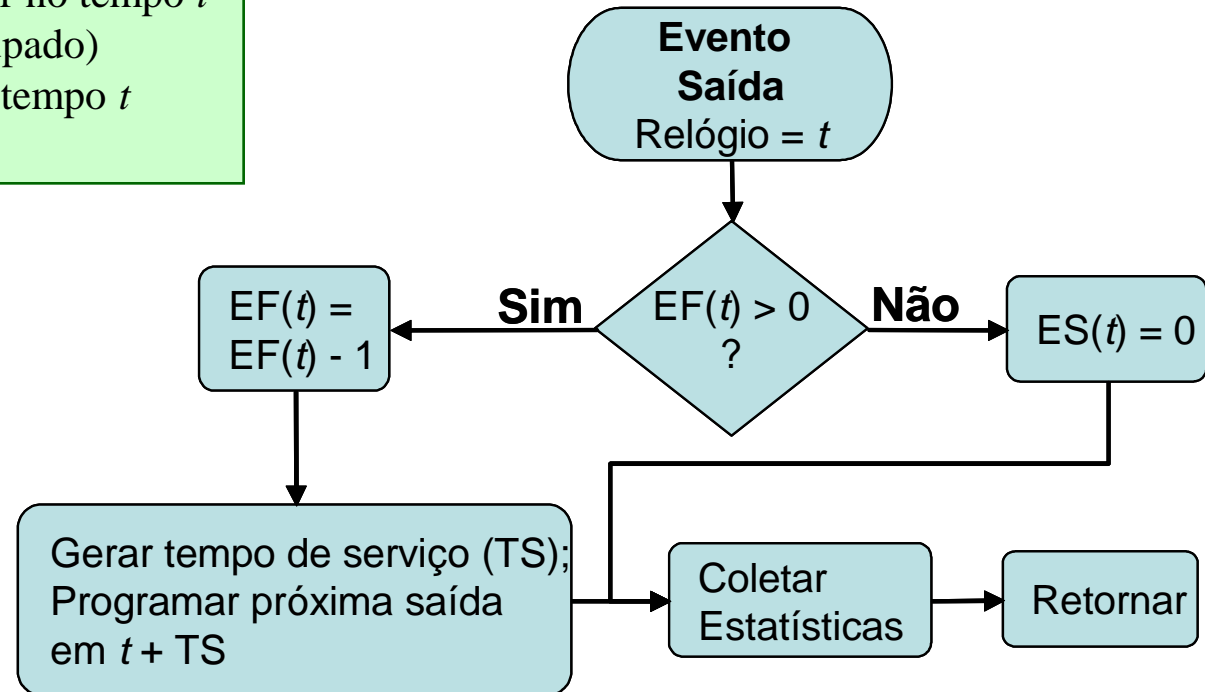
# Exemplo de um Evento Chegada

$ES(t)$  = Estado do Servidor no tempo  $t$   
(0 = livre, 1 = ocupado)  
 $EF(t)$  = Estado da Fila no tempo  $t$   
(0, 1, 2, 3 ....)

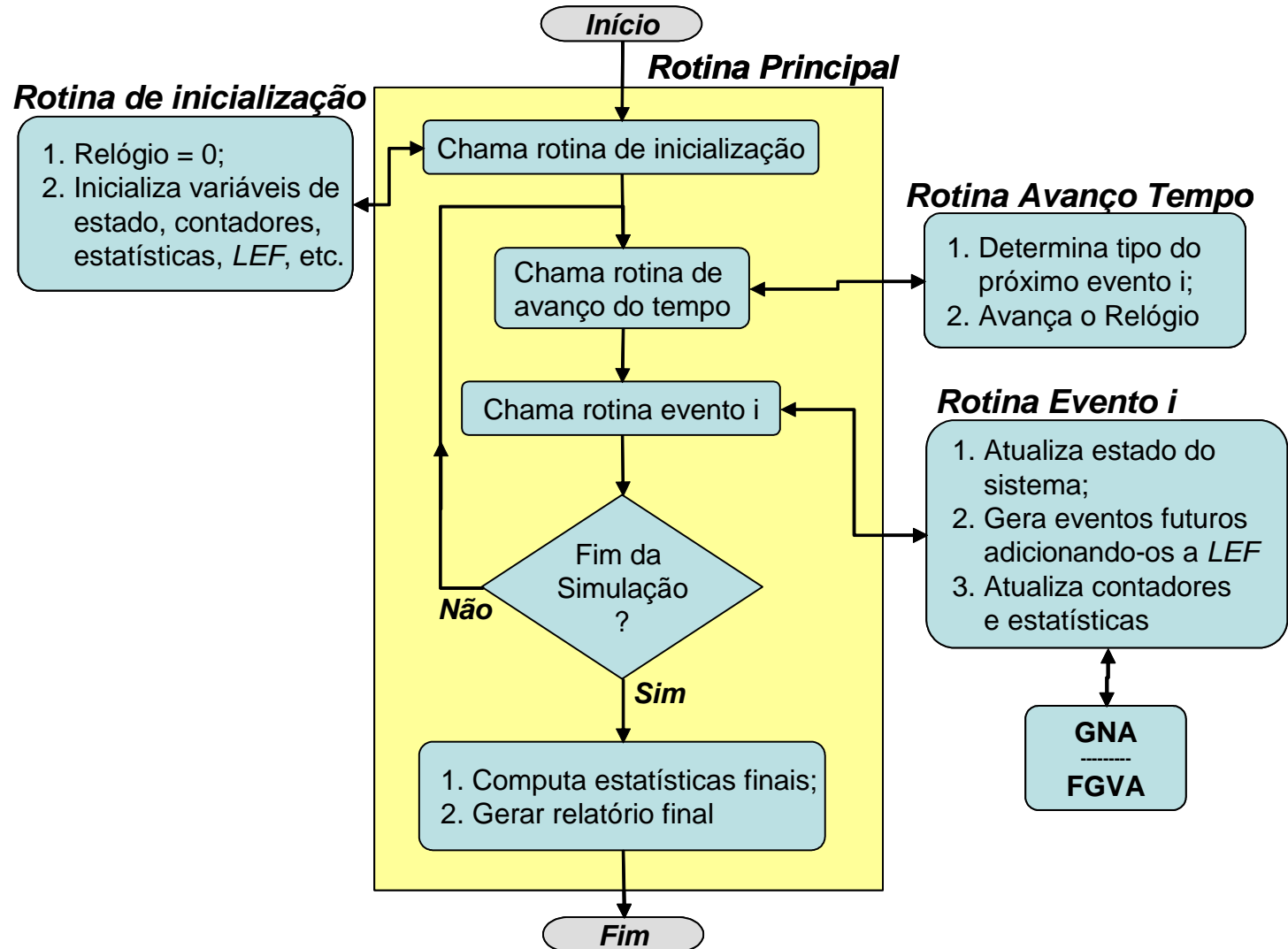


# Evento Saída

$ES(t)$  = Estado do Servidor no tempo  $t$   
(0 = livre, 1 = ocupado)  
 $EF(t)$  = Estado da Fila no tempo  $t$   
(0, 1, 2, 3 ....)



# Fluxograma da Rotina Principal





# Exemplo Aplicação do Algoritmo de Avanço do Tempo

1	2	3	4	5	6	7	8	9	10	11	12
Nº. do Cliente	Tempo no Relógio	Tipo de Evento	Estado Fila (Cliente)	Estado Operador (Cliente)	TC	LEF Lista de Eventos Futuros	Conta Clientes	$\Sigma$ TF	Max. TF	$\Sigma$ TS	Max TS
-	0,0	Início	0	Livre	-	(1; 17,5; C) (-; 240,0; FS)	0	0,00	0,00	0,00	0,00
1	17,5	Chegada	0	Ocupado (1)	17,5	(2; 25,0; C) (1; 29,0; S) (-; 240,0; FS)	0	0,00	0,00	0,00	0,00
2	25,0	Chegada	1 (2)	Ocupado (1)	25,0	(1; 29,0; S) (3; 37,5; C) (-; 240,0; FS)	0	0,00	0,00	0,00	0,00
1	29,0	Saída	0	Ocupado (2)	-	(3; 37,5; C) (2; 41,6; S) (-; 240,0; FS)	1	4,00	4,00	11,5	11,5
3	37,5	Chegada	1 (3)	Ocupado (2)	37,5	(4; 40,0; C) (2; 41,6; S) (-; 240,0; FS)	1	4,00	4,00	11,5	11,5
4	40,0	Chegada	2 (3, 4)	Ocupado (2)	40,0	(2; 41,6; S) (5; 42,5; C) (-; 240,0; FS)	1	6,50	4,00	11,5	11,5
2	41,6	Saída	1 (4)	Ocupado (3)	-	(5; 42,5; C) (3; 53,6; S) (-; 240,0; FS)	2	8,10	4,1	27,1	16,6
...	...	...	...	...	...	...	...	...	...	...	...

# Cronologia dos Eventos

Cliente	Tempo desde a última chegada (minutos)	Tempo de chegada no relógio	Tempo do Serviço (minutos)	Tempo de início do serviço no relógio	Tempo do cliente na fila (minutos)	Tempo final do serviço no relógio
1	17,5	17,5	11,5	17,5	0,0	29,0
2	7,5	25,0	12,6	29,0	4,0	41,6
3	12,5	37,5	12,0	41,6	4,1	53,6
4	2,5	40,0	11,5	53,6	13,6	65,1
5	2,5	42,5	12,0	65,1	22,6	77,1
6	2,5	45,0	10,4	77,1	32,1	87,5
7	2,5	47,5	11,5	87,5	40,0	99,0
8	37,5	85,0	13,1	99,0	14,0	112,1
9	17,5	102,5	10,4	112,1	9,6	122,5
10	17,5	120,0	11,5	122,5	2,5	134,0
11	32,5	152,5	11,5	152,5	0,0	164,0
12	37,5	190,0	9,8	190,0	0,0	199,8
13	7,5	197,5	10,9	199,8	2,3	210,7
14	12,5	210,0	11,5	210,7	0,7	222,2
15	12,5	222,5	10,4	222,5	0,0	232,9
			170,6		145,5	

- ◆ As mudanças de estado do sistema acontecem na medida em que os eventos ocorrem.
- ◆ Os eventos ocorrem em pontos discretos no tempo (..17,5; 25,0;..232,9; 240,0).

Evento	Relógio	Carros no Sistema	Estado do Operador
Início	0,0	0	Livre
Chegada	17,5	1	Ocupado
Chegada	25,0	2	Ocupado
Saída	29,0	1	Ocupado
Chegada	37,5	2	Ocupado
Chegada	40,0	3	Ocupado
Saída	41,6	2	Ocupado
Chegada	42,5	3	Ocupado
Chegada	45,0	4	Ocupado
Chegada	47,5	5	Ocupado
Saída	53,6	4	Ocupado
Saída	65,1	3	Ocupado
Saída	77,1	2	Ocupado
Chegada	85,0	3	Ocupado
Saída	87,5	2	Ocupado
Saída	99,0	1	Ocupado
Chegada	102,5	2	Ocupado
Saída	112,1	1	Ocupado
Chegada	120,0	2	Ocupado
Saída	122,5	1	Ocupado
Saída	134,0	0	Livre
Chegada	152,5	1	Ocupado
Saída	164,0	0	Livre
Chegada	190,0	1	Ocupado
Chegada	197,5	2	Ocupado
Saída	199,8	1	Ocupado
Chegada	210,0	2	Ocupado
Saída	210,7	1	Ocupado
Saída	222,2	0	Livre
Chegada	222,5	1	Ocupado
Saída	232,9	0	Livre
Fim	240,0	0	Livre