

Terminologia Básica

AULA 03

Terminologia Básica

- **Variáveis de Estado**
- **Eventos**
- **Entidades e Atributos**
- **Recursos e Filas de Recursos**
- **Atividades e Períodos de Espera**
- **Tempo (Real) Simulado e Tempo de Simulação**
- **Mecanismos de Avanço do Tempo**

Variáveis de Estado

- As variáveis cujos valores definem o estado de um sistema em um tempo t são conhecidas como **variáveis de estado**.
- Constituem o conjunto de informações necessárias à compreensão do que está ocorrendo no sistema, num determinado instante no tempo, com relação aos objetos de estudo.
- É o que está ocorrendo no sistema em um nível suficiente para atender a uma determinada saída, em um dado ponto no tempo.
- Exemplos:
 - o estado de uma máquina, isto é, ocupada ou livre;
 - número de clientes esperando na fila de um caixa;
 - número de jobs aguardando na fila da CPU, número de jobs já atendidos, etc.

Eventos

- A ocorrência de uma mudança de estado em um sistema é chamada de **evento**.
- Nos exemplos anteriormente citados, os quais representam sistemas com um único servidor e uma única fila de espera por serviço, podemos considerar três tipos de eventos:
- a chegada de peças, clientes ou jobs no sistema,
 - o início do processamento pela máquina, caixa ou CPU e,
 - a saída de peças, clientes ou jobs dos sistemas.

Eventos Incondicionais

- Sua execução depende unicamente do tempo de simulação.

Eventos Condicionais

- Dependem de outras condições além do tempo.
- Geralmente estão relacionadas com o **status do sistema** como, por exemplo, a disponibilidade de um recurso, a espera por outras entidades, etc..

Entidades

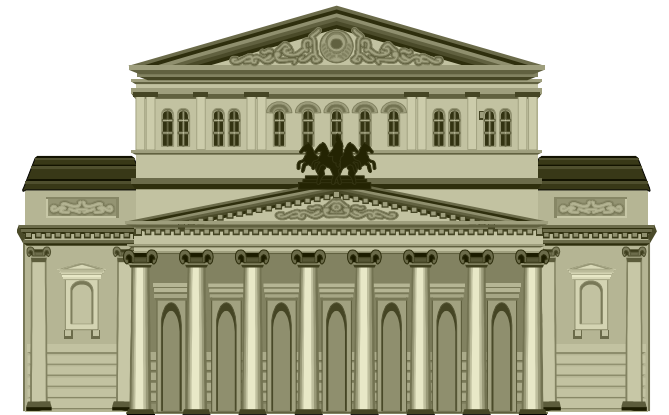
- **Entidades** representam objetos e necessitam definições claras e explícitas.
- Caracterizadas e identificadas através dos seus atributos.
- Podem ser dinâmicas, movendo-se através do sistema, ou estáticas, servindo outras entidades.
- Exemplos de entidades **dinâmicas**:
 - peças (que se movem pela fábrica);
 - clientes chegando e saindo da fila do caixa no supermercado;
 - os jobs que chegam e saem da CPU após processados.
- Exemplos de entidades **estáticas**:
 - a máquina, o caixa e a CPU.

Entidades

- Interagem com recursos, participando de atividades de acordo com condições que determinam a sequência das interações.
- Estas interações estão relacionadas ou criam eventos, os quais alteram o estado do sistema.
- Uma entidade pode estar no:
 - **estado passivo**, ex: esperando numa fila ou
 - **estado ativo**, ex: ao se engajar com outras entidades na execução de uma tarefa (atividade).

Exemplo: Teatro

- Um teatro possui um funcionário para a venda de ingressos. A venda pode ser realizada na bilheteria do teatro ou através de um número telefônico. A central telefônica tem capacidade de manter as ligações em fila.



Exemplo: Teatro

○ Entidades:

- um funcionário
- clientes na bilheteria
- chamadas telefônicas



at.balcão



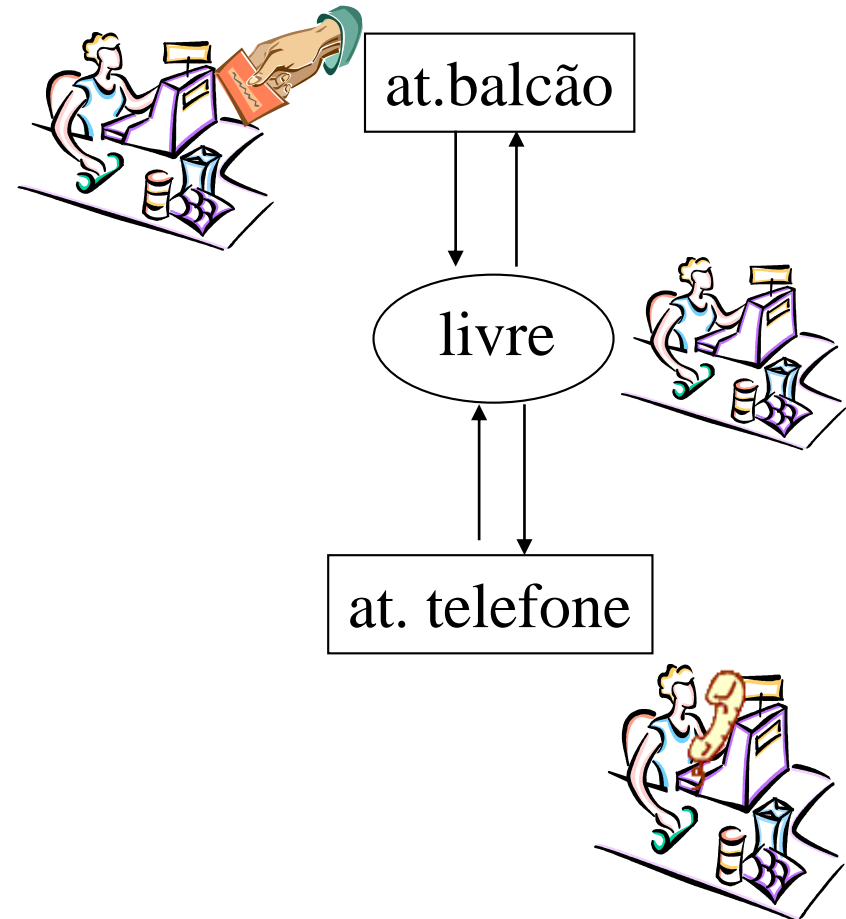
clientes



chamadas

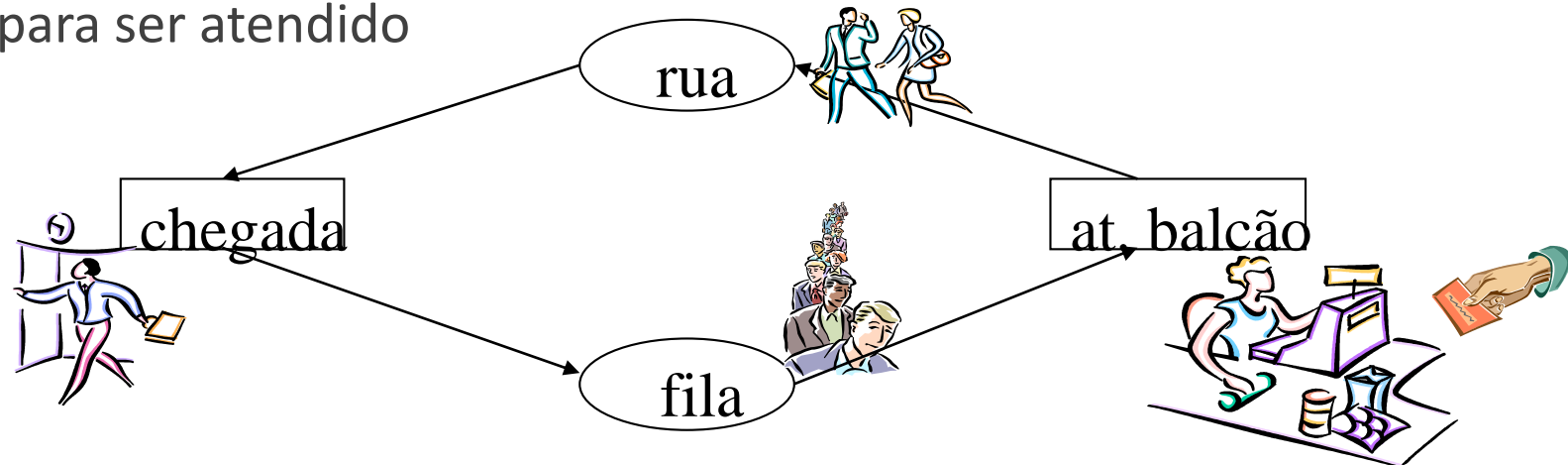
Exemplo: Teatro

- Diagramas:
 - Funcionário:
 - Estados Ativos:
 - at. balcão: atendendo a bilheteria
 - at. telefone: atendendo um telefonema
 - Estados Passivos:
 - livre: funcionário sem atividade



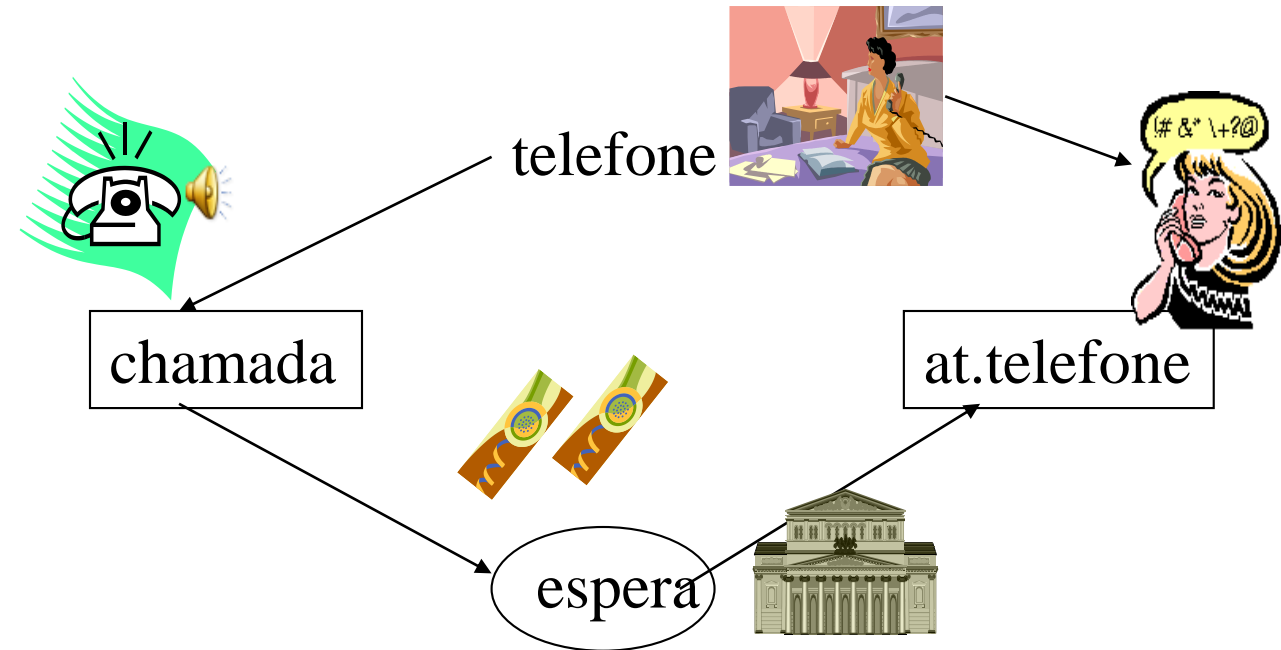
Exemplo: Teatro

- Clientes:
 - Estados Ativos:
 - chegada: cliente chega à bilheteria
 - at. balcão: cliente é atendido
 - Estados Passivos:
 - rua: cliente está inicialmente fora do teatro
 - fila: cliente esperando para ser atendido



Exemplo: Teatro

- Chamadas telefônicas:
 - Estados Ativos:
 - at. telefone: chamada sendo atendida
 - chamada: chamada telefônica chega ao teatro
 - Estados Passivos:
 - telefone: ambiente de origem do telefonema
 - espera: fila de espera da central telefônica



Atributos

- As **características próprias das entidades**, isto é, àquelas que as definem totalmente, são chamadas de **atributos**.
- Várias entidades semelhantes possuem os mesmos atributos. Os valores dos atributos é que as diferenciam entre si.
- Por exemplo, as características de diferentes peças (entidades) produzidas numa máquina podem ser explicitamente individualizadas por um simples atributo tais como um código, número ou tipo.
- O mesmo poderá ocorrer com as entidades clientes ou jobs dos dois outros sistemas exemplificados.

Recursos

- Um recurso é uma **entidade estática que fornece serviços às entidades dinâmicas**.
- Um recurso pode ter a capacidade de servir uma ou mais entidades dinâmicas ao mesmo tempo, operando como um servidor paralelo.
- É possível que uma entidade dinâmica opere com mais de uma unidade do recurso ao mesmo tempo, ou com diferentes recursos ao mesmo tempo.
- Um recurso pode ter vários estados. Os mais comuns são: *ocupado e livre*. Outros podem ser definidos tais como: *bloqueado, falhado, indisponível, etc..*

Recursos

- Uma **entidade dinâmica** pede uma ou mais unidades de um recurso.
 - Se negado, a entidade entra (aguarda) numa *fila de espera*, ou toma outra ação (ex: dirigida a outro recurso ou sai do sistema).
 - Se conseguir o recurso a entidade permanece com ele um tempo (tempo de processamento) e depois o libera.

Filas dos Recursos

- Se uma entidade dinâmica não puder apoderar-se do recurso solicitado, ela deverá **aguardar** pelo mesmo em uma **fila**.
- O **processamento de uma fila**, isto é, a forma como a mesma será gerenciada depende, fundamentalmente, das políticas operacionais adotadas no sistema ou no modelo que o representa.
- A **política de tratamento de fila** mais comum é a FIFO (First In, First Out), onde o primeiro a chegar na fila será o primeiro a ser atendido pelo recurso.
- Se puder **capturar o recurso**, a entidade dinâmica o reterá por um tempo (tempo de processamento) liberando-o a seguir.

Atividades

- Em simulação uma *atividade* corresponde a um período de tempo pré-determinado.
- Uma vez iniciada, seu final pode ser programado.
- A duração de uma atividade não é, necessariamente, uma constante, podendo ser o resultado:
 - de uma expressão matemática;
 - de um valor aleatório com base em uma distribuição de probabilidades;
 - de uma leitura em um arquivo ou fonte externa;
 - ou ser dependente do próprio estado do sistema.

Exemplos de Atividades

- O tempo necessário a um serviço poderá durar 5 minutos para cada entidade, ou 5 minutos para a entidade A e 8 minutos para a B;
- poderá ser aleatório, de acordo com uma distribuição normal de média 10 minutos e desvio-padrão 1 minuto;
- poderá ser dependente do relógio da simulação, isto é, de 5 minutos se o tempo de simulação se encontrar entre o tempo simulado 0 e tempo 120 min, e de 3 minutos se o tempo de simulação for maior do que 120 minutos;
- poderá depender do estado do sistema. Por exemplo: o tempo de serviço depende do tamanho da fila diante de um servidor (caixa).
- Se a fila for grande, maior que cinco entidades, o caixa é levado a agir mais rápido, com média de 3 minutos por entidade.
 - Se no entanto o número de entidades na fila for pequeno (menos do que cinco) ele relaxa e passa a atender com média de 4 minutos por entidade.

Esperas

- Ao contrário de uma atividade, uma **espera** é um período de tempo sobre o qual não temos controle.
- Uma vez iniciado, não podemos programar seu fim.
- Um caso típico, são as esperas causadas por eventos inesperados.
- Todo início e final de uma atividade ou período de espera, é causado por um evento.

Exemplos de Esperas

- Uma entidade entra em uma fila de espera por um recurso. O tempo que a mesma ficará retida na fila dependerá da soma dos tempos de processamento das outras entidades que se encontram na fila ou em processo. Se a forma de gerenciamento da fila for FIFO, pode-se calcular esta espera.
- No entanto, se outra forma de gerenciamento for adotada, com base, digamos, em um atributo de prioridade, um evento inesperado, como a chegada na fila de uma entidade com maior prioridade, torna tal controle quase impossível.
- Outro caso de um evento inesperado seria a indisponibilidade, por tempo indeterminado, do recurso devido a uma quebra do mesmo.

Tempo (Real) Simulado e Tempo de Simulação

- Deve-se tomar certas precauções sobre a relação entre o **tempo** (do sistema real) **simulado** e o **tempo de simulação** (tempo necessário a execução de um experimento no computador).
 - Para certos sistemas, o tempo de simulação pode ser muito maior que o tempo simulado. Por exemplo, na simulação de um modelo de uma rede de computadores, as unidades de tempo admitidas são da ordem de milisegundos.
 - Se no modelo o número de entidades e o número de processos que estas devem ser submetidas for grande (milhares de pacotes sendo roteados, por exemplo), o tempo de CPU dedicado a este processamento poderá ser razoável (dependendo da CPU).
 - Desta forma, para simular, digamos 15 segundos de funcionamento deste sistema, é possível que se gaste dezenas de minutos de tempo de computador.
- **É sempre recomendável estar atento a detalhes desta natureza quando se lida com simulação**

Tipos de Eventos

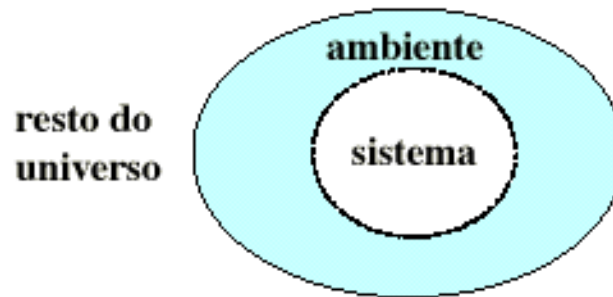
- Eventos que **alteram** valores de **atributos** de entidades.
 - mudança de sinal em um semáforo
- Eventos que **iniciam ou terminam atividades**.
 - cliente começa a ser atendido pelo caixa
- Eventos que **criam ou destroem** uma entidade temporária.
 - cliente chega ao supermercado
 - cliente deixa o supermercado

Tipos de Eventos

- Eventos que adicionam ou removem entidades de conjuntos.
 - cliente entra na fila do caixa
 - cliente sai da fila do caixa
- Obs.: um evento pode ser de vários tipos ao mesmo tempo.

Sistema e Ambiente

- Sistema fechado
 - modelo engloba o sistema e o ambiente.
 - modelo explica todo o universo de interesse.
 - logo, **eventos** são consequências de **eventos anteriores** produzidos pelo **próprio modelo**



Sistema e Ambiente

- Sistema aberto
 - modelo não explica o que ocorre no ambiente.
 - ambiente se reflete no sistema através de eventos exógenos.

Sistemas e Ambiente

- Eventos exógenos
 - modelam atividades não controladas pelo modelo.
 - cliente entra no supermercado
- Eventos endógenos
 - definidos internamente pelo modelo como consequência de eventos anteriores.
 - entrada de cliente na fila do caixa, saída de cliente do supermercado

Geração de Eventos

- Eventos exógenos podem ter geração:
 - **prévia à experimentação**
 - eventos são definidos durante a coleta de dados.
 - armazenamento de um volume muito grande de informações.
 - **iterativa**
 - eventos gerados durante a simulação.

Geração de Eventos Exógenos

- Geração prévia à experimentação.
- programa de simulação acessa os dados à medida que o tempo do modelo é incrementado.
- utilizado quando eventos não seguem alguma regra ou em simulação determinística.
- instruções de uma máquina

Geração de Eventos Exógenos

- Geração iterativa
 - cada evento prepara a geração do evento seguinte.
 - utilizado quando eventos seguem uma regra conhecida, ou seguem uma distribuição de probabilidade (simulação estocástica).
 - eventos não são armazenados previamente.
 - simulação do supermercado, eventos são chegadas de clientes, tempo entre clientes segue distribuição de probabilidade, cada evento calcula o intervalo de tempo até o evento seguinte.

Métodos de Modelagem de Simulação e Funcionamento de Programa de Simulação Orientado a Eventos

Métodos de Modelagem

- *World View* (visão da realidade ou do mundo) é um termo muito utilizado para designar especificamente que abordagem está sendo adotada.
- A visão da realidade sob o ponto de vista da simulação pode ser expressa da seguinte forma [Pegden 90]:
 - A realidade consiste numa série de entidades ou transações que fluem através do sistema. Tais entidades, são descritas, caracterizadas e identificadas por seus atributos.
 - As entidades interagem com recursos e outras facilidades participando de atividades de acordo com certas condições, as quais determinam a sequência das interações.
 - Estas interações estão relacionadas ou criam eventos os quais, alteram o estado do sistema.

Métodos de Modelagem

- Dentro desta *visão* da realidade existem, basicamente, três diferentes métodos de modelagem, os quais dependem da linguagem de simulação empregada para serem utilizados:
 - a modelagem por eventos;
 - a modelagem por atividades;
 - a modelagem por processos.
- Em todas as abordagens, quando o próximo evento é selecionado para ser processado, uma parte da lógica do modelo é executada ocorrendo assim mudanças no estado do sistema.

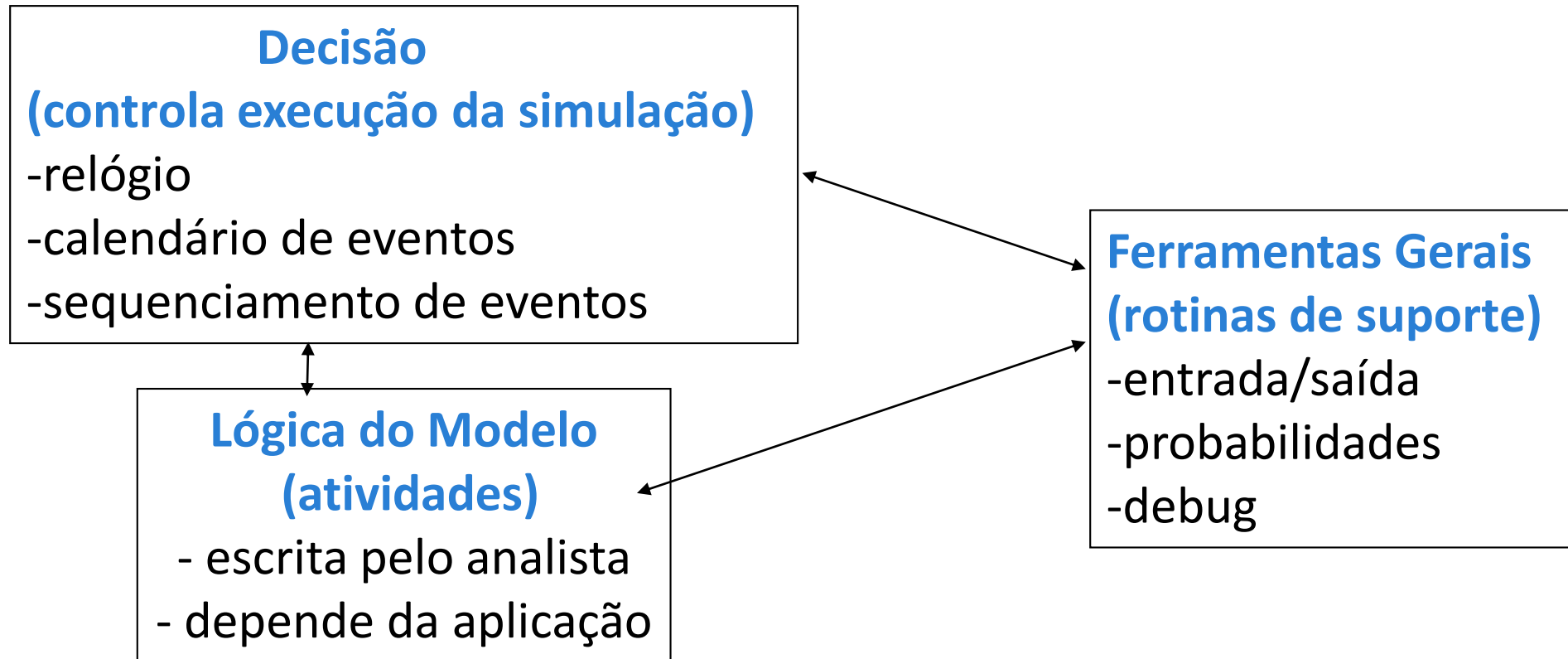
Métodos de Modelagem

- As diferenças básicas entre os métodos encontram-se, fundamentalmente, na forma com que o *próximo evento* é programado para ser processado, o que depende do mecanismo de avanço do tempo adotado.

Método de Modelagem

- Uma implementação de simulação discreta apresenta, basicamente, 3 grandes blocos:
 - **Decisão:** controla execução da simulação.
 - **Ferramentas Gerais:** rotinas de suporte.
 - **Lógica do Modelo:** expressa as atividades nas quais as entidades do sistema estão engajadas.

Métodos de Modelagem



Cada abordagem diferente implica em blocos de Decisão e Lógica diferentes

Métodos de Modelagem

- Um sistema pode ser modelado utilizando **qualquer uma** das três abordagens.
- Uma abordagem pode ser **mais conveniente** que outra dependendo do sistema a ser modelado.

Modelagem por Eventos

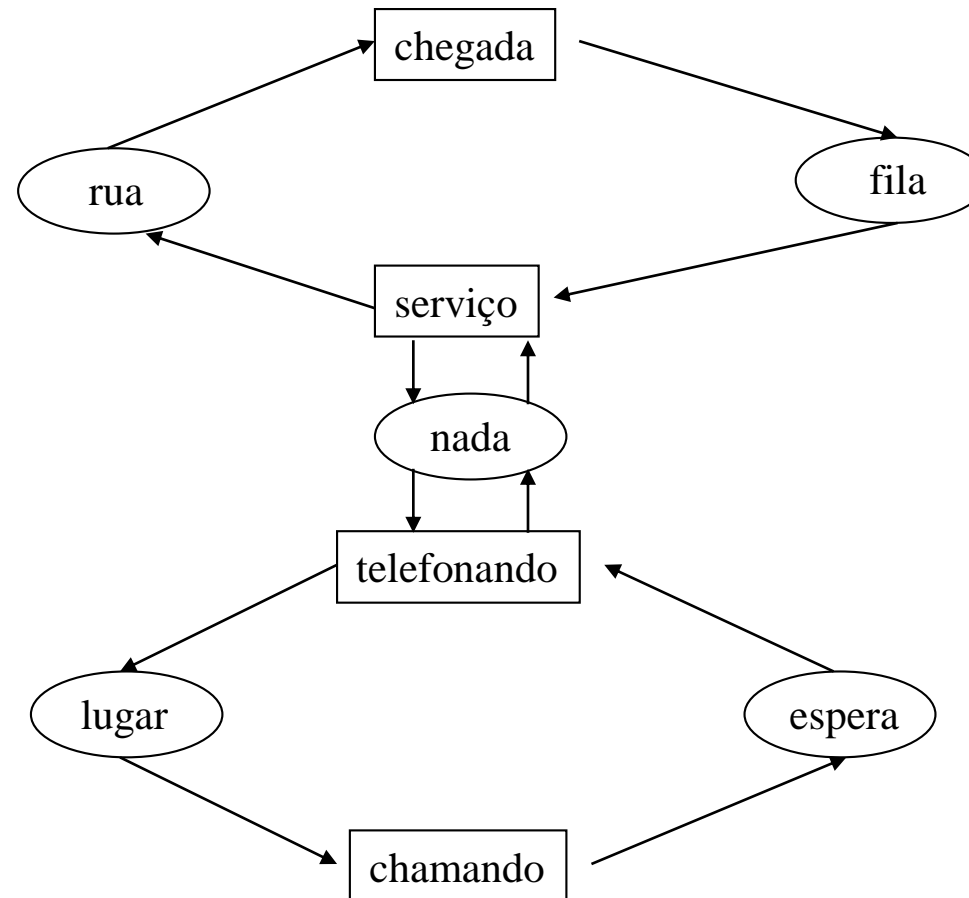
○ *Abordagem por Eventos*

- 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 relacionadas a ocorrência dos eventos.
- O processo de simulação evolui ao longo do tempo pela execução dos eventos selecionados de uma ***pilha de eventos***, escolhendo sempre aquele com o tempo (determinado) mais próximo do tempo corrente de simulação.

Exemplo: Teatro

- 4 eventos:
 - Chegada: chegada de um cliente na bilheteria
 - Chamada: chegada de uma chamada telefônica
 - FimDeServiço: fim do atendimento de um cliente na bilheteria
 - FimDeChamada: fim do atendimento de uma chamada telefônica

Exemplo: Teatro



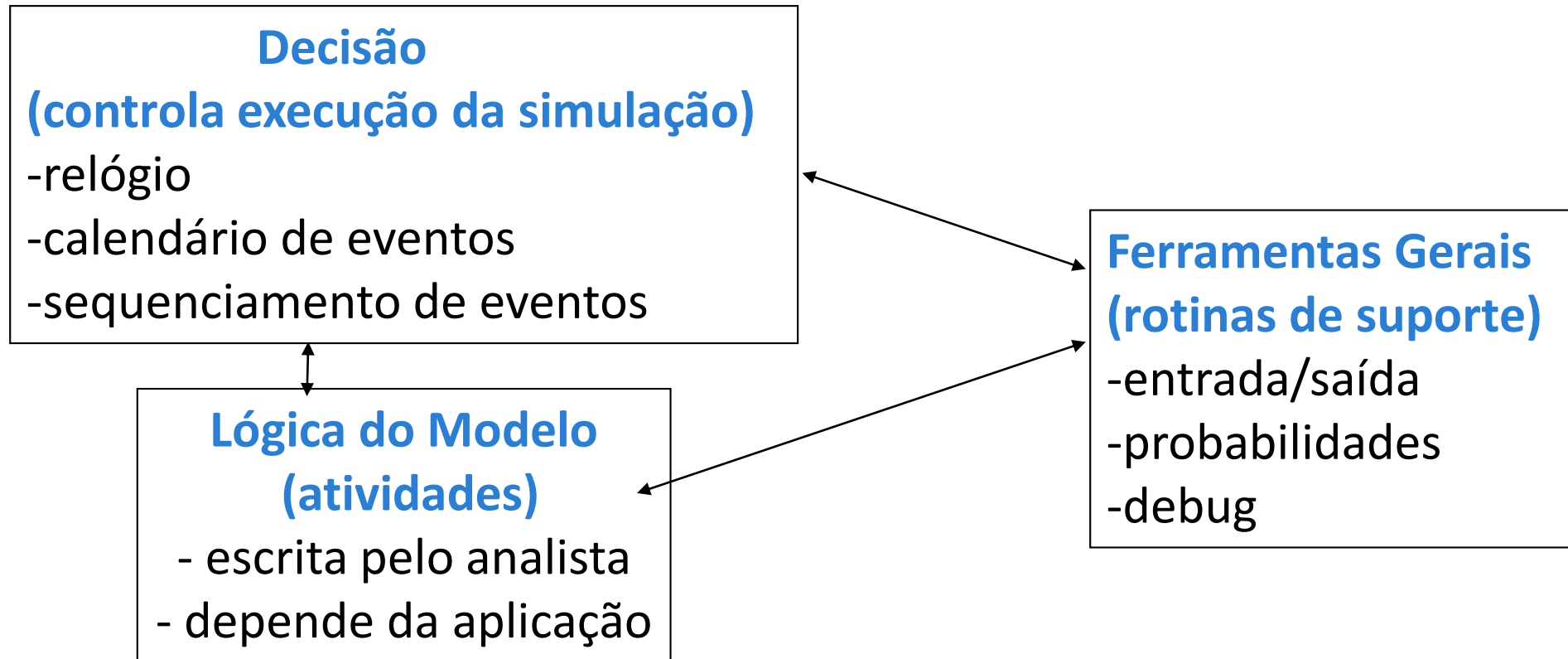
Rotina de Evento

- FimDeServiço
 - libere Cliente
 - se (fila de clientes não vazia) então
 - retire próximo Cliente da fila
 - calcule TempoDeServiço
 - agende evento FimDeServiço para ocorrer após TempoDeServiço
 - senão se (fila de telefonemas não vazia) então
 - retire próximo Telefonema da fila
 - calcule TempoDeConversa
 - agende evento FimDeChamada para ocorrer após TempoDeConversa
 - senão libere Funcionário.

Modelagem por Eventos

- Mecanismos de Avanço do Tempo
- Em função da natureza dinâmica dos modelos de simulação discreta é preciso 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*.

Métodos de Modelagem



Cada abordagem diferente implica em blocos de Decisão e Lógica diferentes

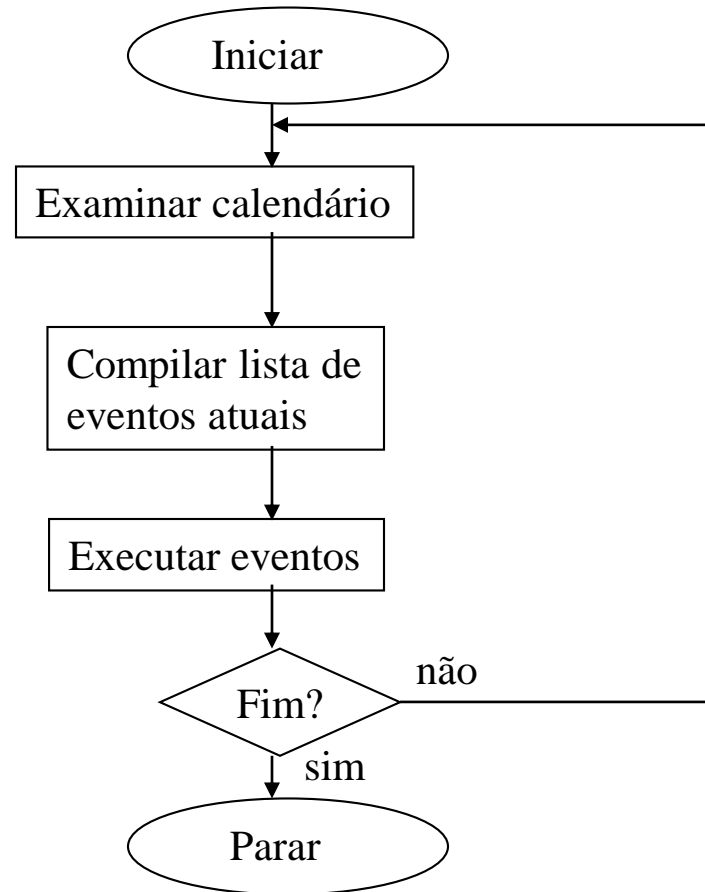
Modelagem por Eventos

- Módulo de Decisão:
 - Realiza o agendamento dos eventos e não se preocupa com o sequenciamento.
 - Sequenciamento é tarefa das rotinas de eventos.
 - Consequência para o processo de simulação: lógica de modelo é mais complexa.

Modelagem por Eventos

- O módulo de decisão divide-se em 2 fases:
 - **Examinar o calendário de eventos** para descobrir qual o próximo evento a ser processado e mover o relógio de simulação para o tempo do evento;
 - **Mover todas as referências para as rotinas de evento** associadas com este tempo para uma lista de eventos atuais.
 - Mantendo o relógio constante, **executar cada rotina de evento da lista** de eventos atuais.

Método por Eventos



Método por Eventos

- Dois mecanismos de avanço do tempo aparecem nas diversas linguagens de simulação:
 - avanço do tempo com incremento fixo;
 - avanço do tempo para o próximo evento.
 - amplamente utilizado tanto por programas comerciais de simulação como por aqueles montados sobre uma linguagem de programação de propósito geral como C, Delphi, VB ou Java.

Modelagem por Eventos

Eficiente em modelos c/ componentes relativamente independentes

- quando as operações sobre as transações se sucedem como numa linha de montagem, sem muita interdependência entre elas.

2) Flexível:

- aplicável a uma gama muito ampla de modelos;
- constitui-se numa boa base para simulações de sistemas específicos.

3) Maior esforço de desenvolvimento

- as rotinas de eventos devem ser escritas com base em linguagens de propósito geral, que requerem habilidade de programação

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 Atividades

- A estratégia de busca do próximo evento da lista é baseada:
 - tanto no tempo programado de ocorrência (tempo de simulação) e
 - em testes condicionais relativos a **quaisquer outras condições favoráveis ao disparo de um evento.**

Modelagem por Atividades

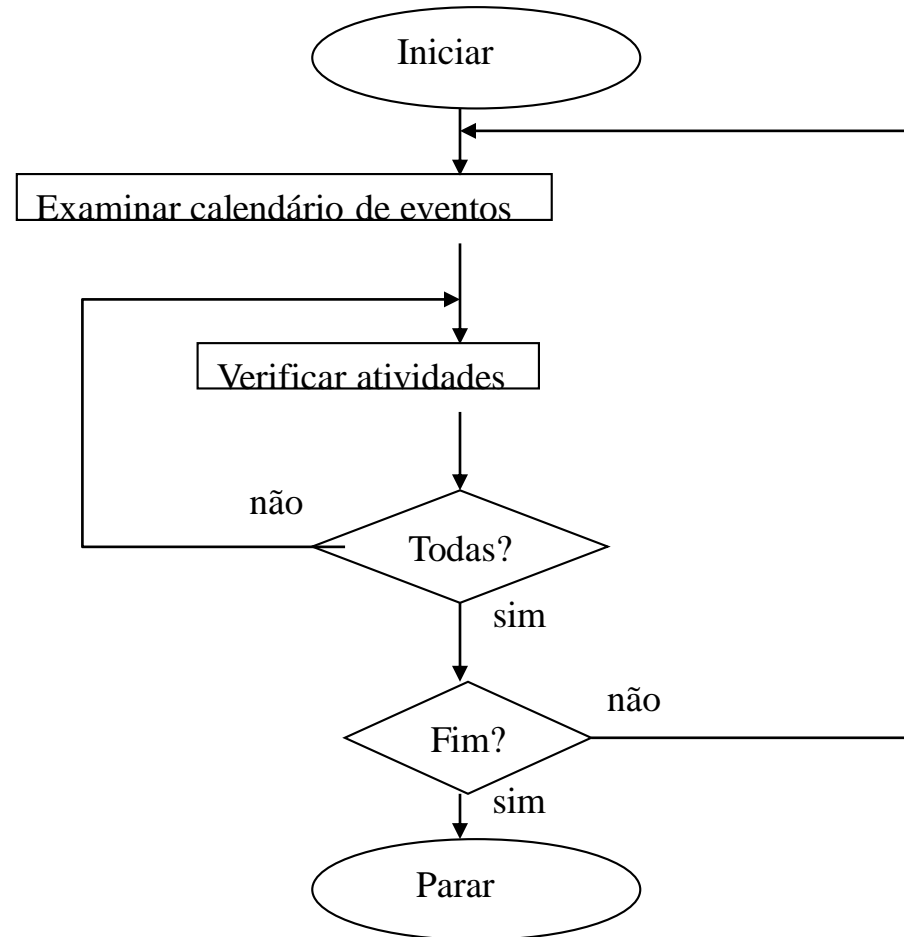
- A **monitoração** de situações de busca de condições de início de eventos é típica da modelagem por atividades.
- Existe prioridade entre atividades.
- Lógica do Modelo
 - Atividades são obtidas do diagrama de ciclos de atividades.
 - Atividades têm a forma de sequências de instruções que são executadas se uma determinada condição é satisfeita.

Modelagem por Atividades

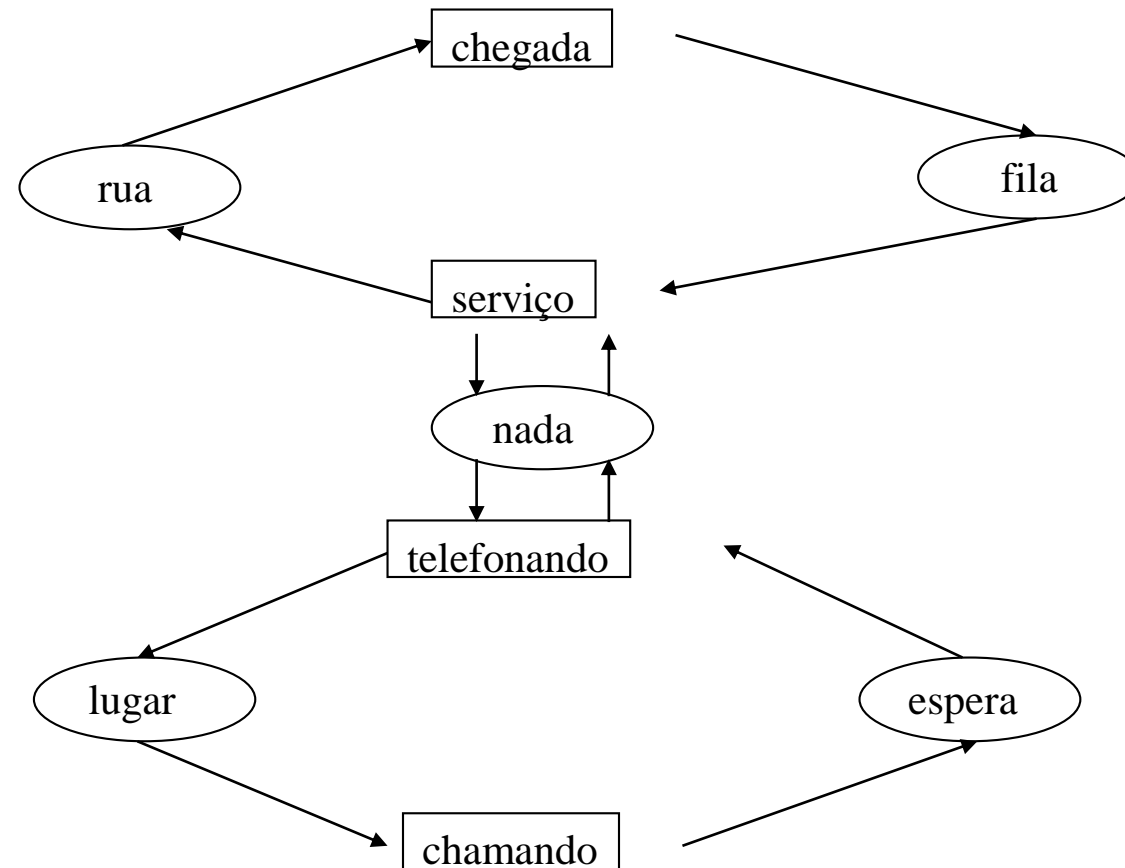
- Módulo de Controle

- Divide-se em duas fases principais:
 - Verificar o calendário de eventos para obter o tempo do próximo evento. Avançar o relógio de simulação para o novo tempo.
 - Verificar todas as atividades e testar a condição de ativação das mesmas. Executar todas as atividades possíveis.
- Simulação mais lenta, pois necessita verificar todas as atividades, mesmo sabendo de antemão que alguma não vai ocorrer.

Modelagem por Atividades



Exemplo: Teatro



Exemplo: Teatro

- chegada de cliente
- início de serviço
- fim de serviço
- chegada de telefonema
- início atendimento de telefone
- fim de atendimento de telefone

Exemplo: Teatro

- Início de serviço
 - se (Funcionário livre) e (fila clientes não vazia) então
 - retire Cliente da fila clientes
 - coloque Funcionário em serviço
 - calcule TempoDeServiço
 - agende atividade FimDeServiço para ocorrer após TempoDeServiço

Exemplo: Teatro

○Chegada

se (tempo agendado da atividade = TempoAtual) então

coloque Cliente na fila clientes

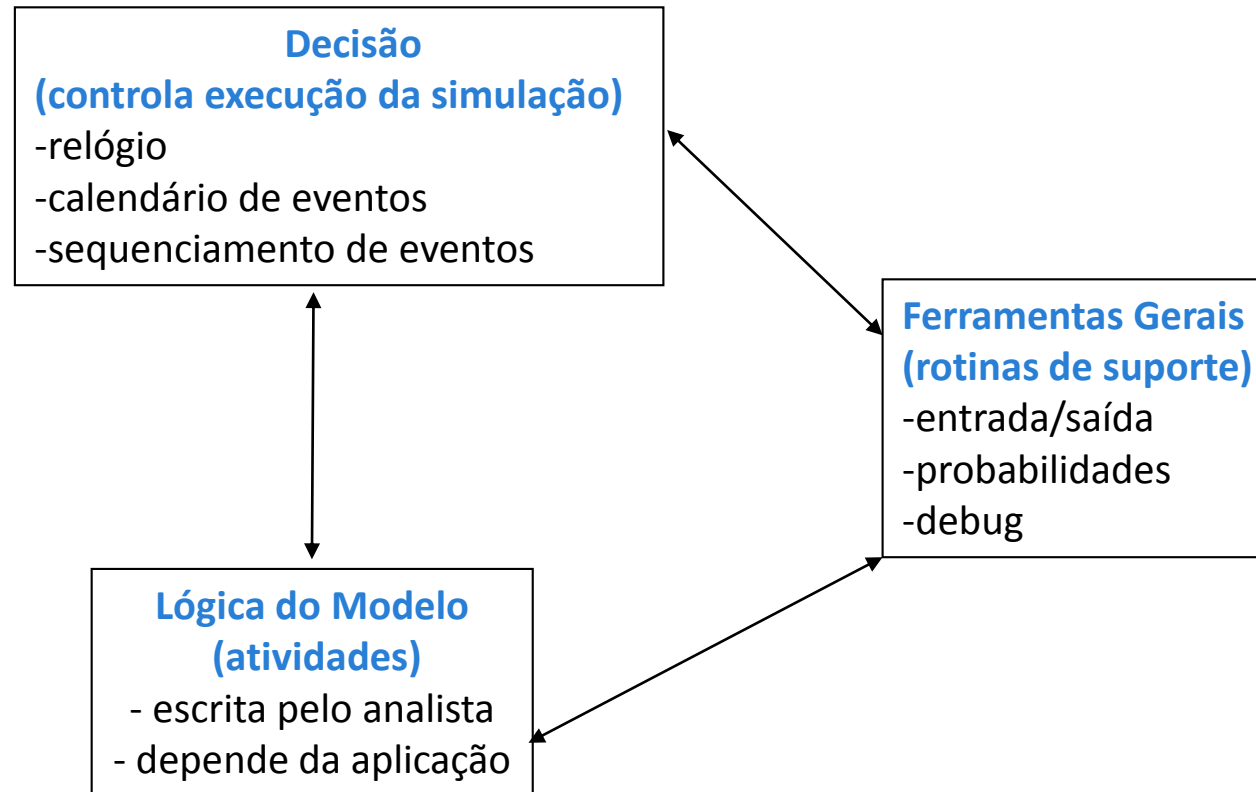
calcule TempoChegada do próximo Cliente

agende atividade Chegada para ocorrer após TempoChegada

Modelagem por Atividades

- Execução eficiente para modelos com componentes fortemente dependentes.
- Falta de encadeamento de partes do modelo, as rotinas das atividades aparecem pulverizadas no modelo;
- Boa parte do trabalho é feita pelo controlador da linguagem: teste das condições; a determinação do fluxo de controle do programa.
- O sequenciamento dos testes relacionados com as atividades deva ser, no entanto, previamente explicitado pelo usuário no interior do programa;

Modelagem por Atividades



Cada abordagem diferente implica em blocos de Decisão e Lógica diferentes

Métodos de Modelagem – Por Processos

- os sistemas são vistos principalmente do ponto de vista das entidades.
- a visão é de um conjunto das entidades dinâmicas de um sistema que se encontram permanentemente competindo entre si pela posse de recursos.
- o programa de computador resultante da modelagem procura, por meio de suas rotinas, imitar o comportamento das entidades em seu fluxo ou movimento pelo sistema, buscando cumprir os processos ou atividades a elas designados.
- o programa monitora cada entidade individualmente. Seu movimento ou sequência de atividades é acompanhado até que, num ponto qualquer, este seja suspenso. No momento em cessa seu movimento, o programa avança o relógio da simulação para o tempo em que outra entidade (a primeira da fila de próximo evento) esteja sendo liberada de uma suspensão e passa a acompanhá-la.
- estes procedimentos se repetem com todas as entidades que estejam no sistema até que estas o deixem, quando realizarem todas as suas atividades.

Métodos de Modelagem – Por Processos

- Processo: sequência de operações pela qual uma entidade deve passar durante seu ciclo de vida.
- Cada classe de entidades tem seu próprio processo.
- A simulação consiste em um conjunto de processos, pelo menos um para cada classe de entidades.

Modelagem por Processos

- 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.

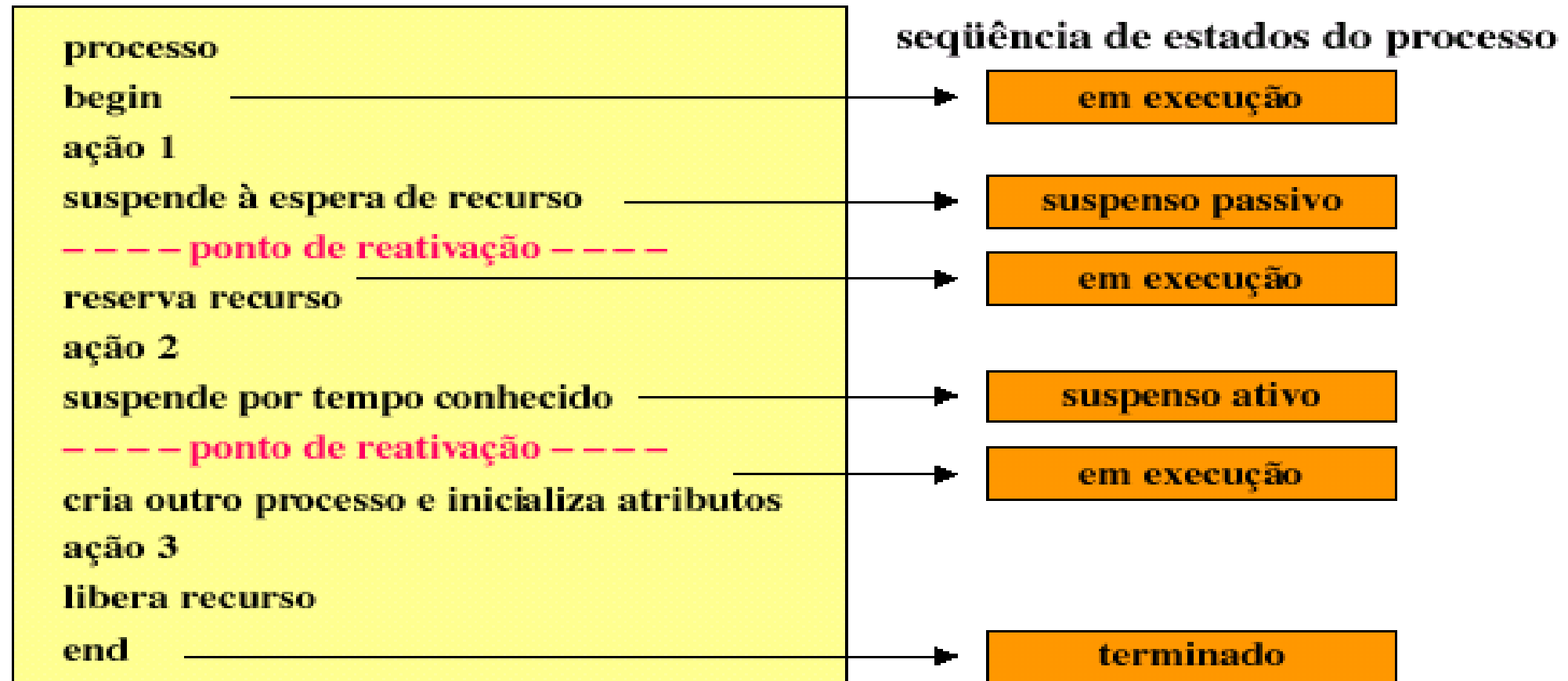
Modelagem por Processos

- O progresso de uma entidade por um processo é regulado por dois tipos de condições:
 - **Bloqueio incondicional**
 - Também chamado de suspensão ativa.
 - Ocorre quando o progresso de uma entidade é bloqueado por um certo período de tempo que, em princípio, pode ser determinado de antemão.
 - Depois de decorrido o período de tempo indicado, a entidade é desbloqueada.
 - Ex.: atendimento de um cliente

Modelagem por Processos

- O progresso de uma entidade por um processo é regulado por dois tipos de condições:
 - **Bloqueio condicional**
 - Também chamada de suspensão passiva.
 - Ocorre quando o progresso de uma entidade é bloqueado devido a condições específicas do modelo simulado.
 - A entidade é desbloqueada no momento em que as condições são satisfeitas.
 - Ex.: cliente permanece na fila até que chegue na primeira posição da fila e o caixa esteja livre.

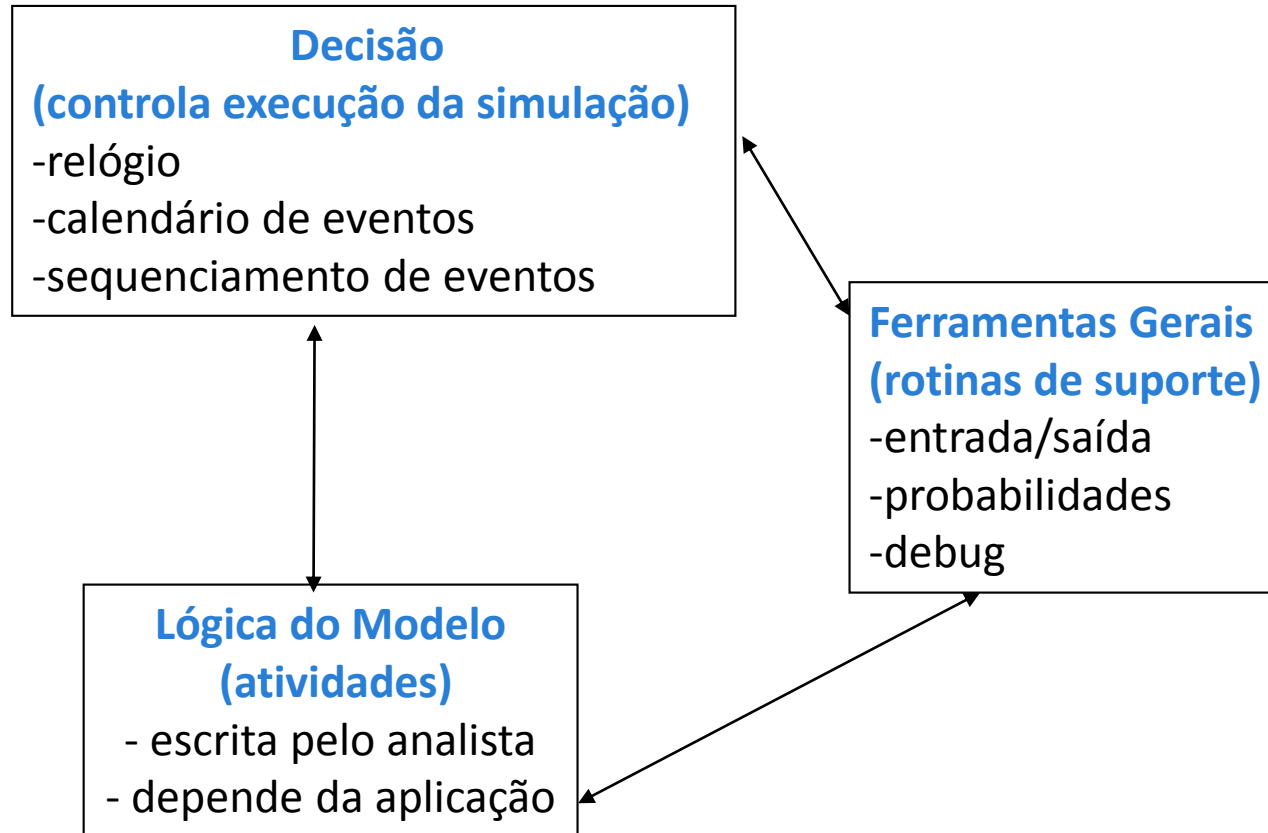
Modelagem por Processos



Modelagem por Processos

- Módulo de decisão
 - Necessita saber onde, em cada processo, uma entidade está (o estado do processo) e necessita também ter a capacidade de parar e reiniciar o movimento de uma entidade pelo seu processo.
 - Logo, deve manter, para cada entidade:
 - tempo de reativação (se conhecido)
 - ponto de reativação (isto é, a posição dentro do processo)

Modelagem por Processos



Cada abordagem diferente implica em blocos de Decisão e Lógica diferentes.

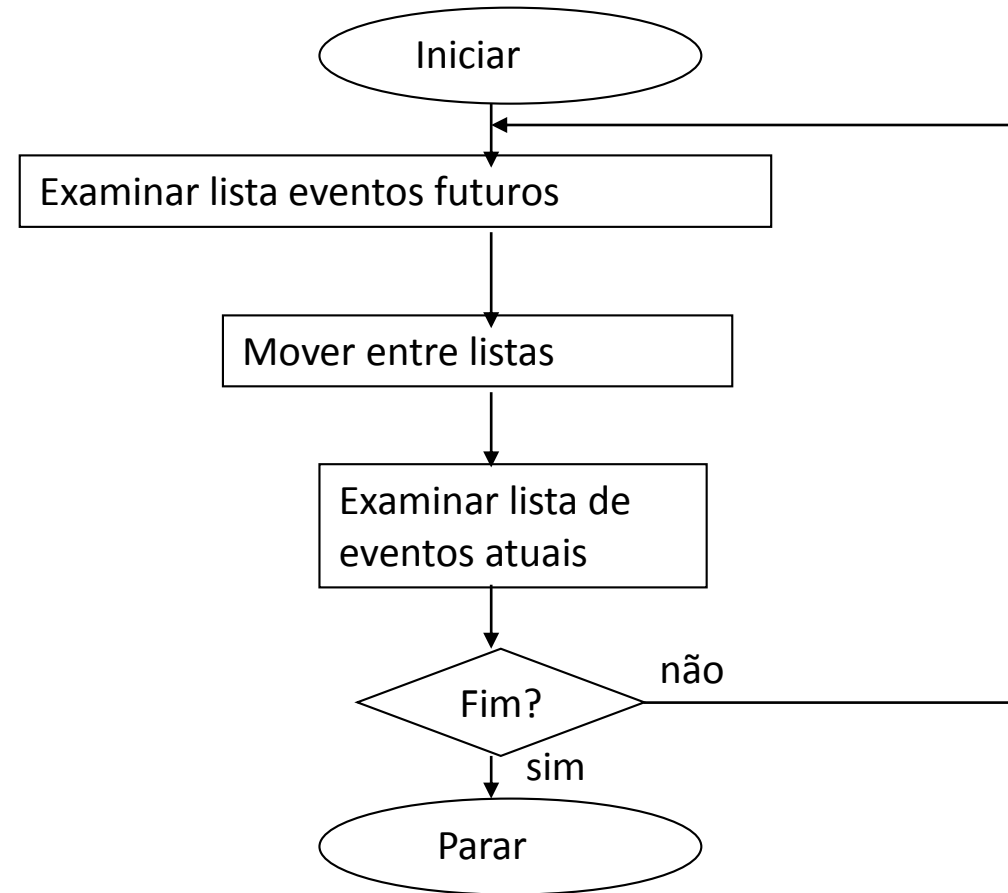
Modelagem por Processos

- O módulo de decisão mantém duas listas com essas informações:
 - lista de eventos futuros: calendário com as informações sobre entidades que sofreram bloqueio incondicional.
 - lista de eventos atuais: contém as informações de dois tipos de entidades
 - entidades bloqueadas incondicionalmente (por certo período de tempo) que atingiram o tempo de desbloqueio,
 - entidades em bloqueio condicional.

Modelagem por Processos

- O módulo de decisão divide-se em três fases:
 - A lista de eventos futuros é utilizada para se determinar o **próximo evento**. O relógio de simulação é atualizado para o novo tempo.
 - As entidades da lista de eventos futuros cujo tempo de reativação for igual ao relógio são **movidas para a lista de eventos atuais**.
 - Cada entidade da lista de eventos atuais é **movimentada pelo seu processo** se as condições permitirem, até o ponto de terminar o processo ou serem bloqueadas novamente.

Modelagem por Processos



Modelagem por Processos

A representação do modelo pelo simulador é a mais "natural" ou "próxima do problema":

- o fluxo lógico é similar ao processo no sistema real;
- especialmente aplicáveis para sistemas com padrões de comportamento previsíveis (sequência de ações conhecidas e de repetição regular).

Modelagem por Processos

Desenvolvimento/alteração do modelo são mais diretas:

- redução de declarações repetitivas nas rotinas;
- linguagens baseadas nesta estratégia são mais fáceis de serem aprendidas;
- os programas produzidos tendem a ser mais compactos;
- os modelos são legíveis
- pode ter tempo de execução um tanto ineficiente.

Funcionamento de um programa de Simulação

- A simulação discreta envolve a modelagem de **sistemas dinâmicos** nos quais as **mudanças de estado** acontecem em pontos discretos ao longo do tempo, mais precisamente, no exato momento da ocorrência de **eventos**.
- O processo de simulação produz uma sequência de ***imagens ou instantâneos*** que representam 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.

Imagem de um Sistema no Tempo t

- Uma imagem de um sistema num dado momento deve incluir, pelo menos, os seguintes elementos:
 - **o estado do sistema no tempo t ;**
 - **uma lista das atividades em progresso, com seus respectivos tempos de término;**
 - **o estado de todas as entidades presentes no sistema;**
 - **os atuais membros de todas as listas ou conjuntos (coleção temporária ou permanente de entidades, como por exemplo filas de espera;**
 - **os valores atualizados de estatísticas acumuladas e de contadores, os quais serão utilizados para a produção de relatórios sumários ao final da simulação.**

Mudanças de Estado/Eventos

- De maneira simplificada, considera-se que no modelo a ser simulado as mudanças de estado se sucedem devido a ocorrência de apenas três tipos de eventos:
 - **Evento Chegada;**
 - **Evento Saída;**
 - **Evento Fim da Simulação.**

Evento Chegada

- Diante da chegada de uma entidade no sistema, as seguintes ações devem ser realizadas pelo programa de simulação:
 - **Guardar o tempo de ocorrência da chegada da entidade em um atributo especialmente designado para tal. Este valor servirá para que no momento de sua saída, o tempo de permanência da entidade no sistema possa ser calculado;**
 - **Verificar se o recurso com o qual a entidade realizará uma atividade está livre. Se estiver, o tempo de permanência da entidade na fila deste recurso será zero. Neste caso, trocar o estado do recurso para *ocupado* e programar o final da atividade no calendário de eventos, somando o tempo atual do relógio ao tempo de serviço sorteado;**

Evento Chegada

- Diante da chegada de uma entidade no sistema, as seguintes ações devem ser realizadas pelo programa de simulação:
 - **Se o recurso estiver ocupado, a entidade deve ser colocada no final da fila do recurso. A variável tamanho da fila deve ser incrementada em um.**
 - Programar a chegada de uma nova entidade no sistema, no calendário de eventos, somando o tempo do relógio com tempo entre chegadas sorteado para a próxima entidade.

Evento Saída

- No momento que uma entidade completar sua atividade junto a um recurso, as seguintes ações devem ser realizadas pelo programa de simulação:
 - Incrementar o contador de entidades servidas;
 - Computar e armazenar o tempo de permanência da entidade no sistema fazendo tempo do relógio menos o valor do atributo tempo de chegada da entidade.
 - Se houver alguma entidade na fila de espera, retirar a primeira entidade da fila, computar seu tempo de permanência na fila, decrementar a variável tamanho da fila, iniciar o atendimento da nova entidade e programar no calendário de eventos futuros o fim da atividade, somando o tempo de serviço ao tempo atual;
 - Se não houver nenhuma entidade na fila de espera, fazer o estado do recurso igual a *livre*.

Evento Fim da Simulação

- Quando for apropriado, isto é, quando da ocorrência de um evento especial o qual determina o fim da execução da simulação, as seguintes ações devem ser realizadas:
 - **Computar as estatísticas idealizadas para o fim da simulação;**
 - **Compor o relatório final com as estatísticas a ser exibido ao usuário.**

Dinâmica do Sistema

- A definição dos componentes de um modelo permite uma descrição estática do mesmo.
- E como fica o relacionamento dinâmico entre seus componentes?

Dinâmica do Sistema

- De que maneira a **ocorrência** de cada **evento** afeta o **estado do sistema**, os atributos das entidades e o conteúdo dos conjuntos?
- Como as **atividades** são **definidas**, isto é, são determinísticas, probabilísticas ou dependem de outra função matemática qualquer?
- Que **eventos** marcam o **início** ou o **fim** de cada **atividade**?

Dinâmica do Sistema

- Uma atividade pode ser iniciada não importando o estado do sistema ou é condicionada a ele?
 - Exemplo: um carro não pode iniciar a atividade de lavação, se não houver um operador disponível
- Que eventos dão início e fim a cada tipo de espera? Sob que condições inicia e termina um período de espera?

Dinâmica do Sistema

- Qual o estado do sistema no tempo zero?
- Que eventos devem ser gerados no tempo zero para dar início à simulação?
- Como se encerra a execução da rodada de simulação ?

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.
- Muitos algoritmos para a programação de eventos empregam estruturas de dados, tais como listas ou pilhas encadeadas. No entanto, a execução deste método não requer, necessariamente este tipo de estrutura.
- Um dos principais elementos dos algoritmos voltados ao método da programação de eventos é a *calendário de eventos futuros*.

Calendário de Eventos Futuros

- 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. A dinâmica dos acontecimentos é realizada da seguinte maneira:
 - O relógio da simulação é avançado para o valor programado para a ocorrência do evento que se encontra no topo da lista (calendário de eventos).
 - Após cada a realização do evento, este é retirado da lista.
 - A cada vez que um evento é programado, isto é, é incorporado na lista, esta é reordenada. Desta maneira, o evento mais eminente estará sempre no topo da lista e aquele mais distante no tempo estará sempre ocupando a última colocação na lista.

Controle num programa de simulação

| Nº do Cliente | Tempo no Relógio | Tipo de Evento | Estado Fila (Client) | Estado Operad. (Client) | TC | Calendário de Eventos Futuros | Conta Clientes | Σ TF | Max. TF | Σ TS | Max TS |
|---------------|------------------|----------------|----------------------|-------------------------|------|--|----------------|-------------|---------|-------------|--------|
| - | | Início | 0 | Livre | - | (1; 17,5; C) (-; 240,0; FS) | 0 | 0,00 | 0,00 | 0,00 | 0,00 |
| 1 | 17,5 | Chegada | 0 | Ocup. (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) | Ocup. (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 | Ocup. (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) | Ocup. (2) | 37,5 | (4, 40,0; C) (2: 41,6; S) (-; 240,0; FS) | 1 | 4,00 | 4,00 | 11,5 | 11,5 |

Tabela com os elementos de controle de um programa de simulação

Algoritmo de Avanço do Tempo

- A sequência de ações que um simulador precisa realizar para avançar o relógio da simulação e criar uma nova imagem do sistema é baseada em um algoritmo conhecido como:

Algoritmo para Avanço do Tempo com base na Programação de Eventos.

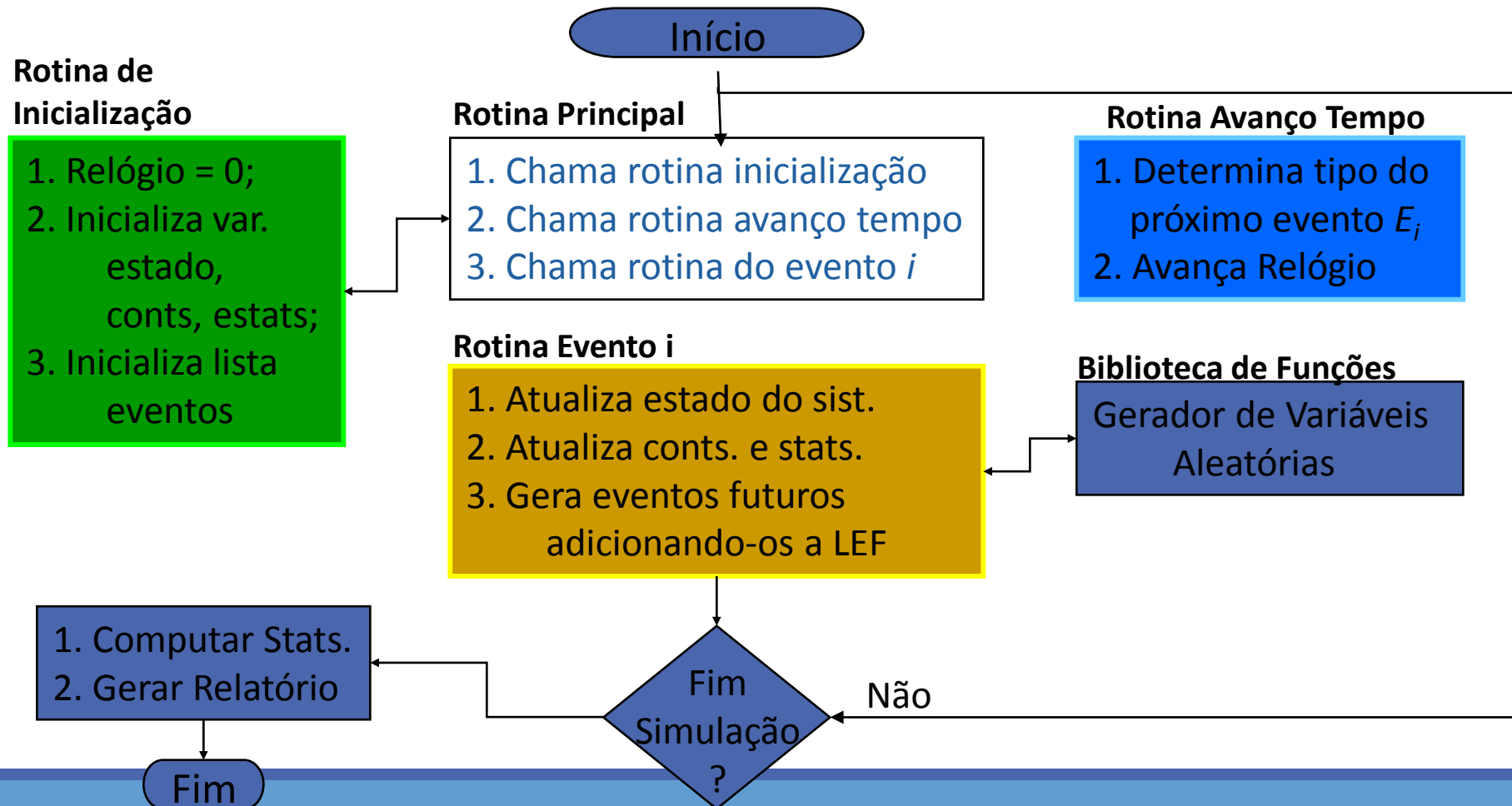
Algoritmo de Avanço do Tempo

- Remova o evento iminente do calendário;
- Avance o RELÓGIO para o tempo do evento iminente;
- Execute o evento iminente atualizando (na medida do necessário): o estado do sistema, os atributos das entidades e os membros de conjuntos;
- Gere futuros eventos (se necessário) e coloque-os no calendário, na posição correta (ordem cronológica);
- Atualize estatísticas acumuladas e contadores.

Programa de Simulação

- Um programa voltado à **simulação de modelos de mudança discreta** é composto por:
 - Rotina Principal;
 - Rotina de inicialização;
 - Rotinas de Eventos (ex: evento chegada e saída);
 - Rotina de Avanço do Tempo;
 - Biblioteca de Funções;
 - Gerador de Relatórios.

Fluxograma da Rotina Principal



Funcionamento de um Programa de Simulação

○ Rotina Principal:

- Elemento central, responsável pela chamada das demais rotinas e pelo controle da simulação;

○ Rotina de Inicialização:

- Define variáveis, atributos, var. de estado, etc. início da simulação.
- Os valores podem ser preestabelecidos ou fornecidos pelo usuário ao início de cada rodada de simulação.

○ Rotina do Evento E_i :

- São rotinas que atualizam o estado do sistema quando um tipo particular evento ocorre.
- Deve existir uma rotina de evento para cada um dos possíveis E_i 's. (Ex. Eventos Chegada e Saída);

Funcionamento de um Programa de Simulação

○ **Rotina de Avanço do Tempo:**

- Determina o próximo evento e avança o RELÓGIO da simulação para o tempo em que o evento deve ocorrer.

○ **Bibliotecas de Funções:**

- Para geração de valores aleatórios, de acordo com as principais distribuições teóricas de probabilidades exigidas pelo modelo durante a simulação.

○ **Gerador de Relatórios:**

- Computa e gera relatórios parciais ou finais sobre as inferências desejadas pelo usuário, a partir das estatísticas e dos contadores definidos no programa.

Calendário de Eventos Futuros

| Num do Cliente | Tempo no relógio | Tipo de Event | EstadFil a Client | Estado Operad | TC | CalenEvents Futurs | Cont Clien | SomaTF | Max TF | SomaTS | Max TS |
|----------------|------------------|---------------|-------------------|---------------|------|--|------------|--------|--------|--------|--------|
| - | | Início | 0 | Livre | - | (1; 17,5; C) (-;240,0;FS) | 0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1 | 17,5 | Cheg. | 0 | Ocup. (1) | 17,5 | (2; 25,0; C) (1; 29,0; S) (-;240,0;FS) | 0 | 0,0 | 0,0 | 0,0 | 0,0 |
| | | | | | | ... | | | | | |

No tempo = 17, 5 ocorre a chegada do primeiro cliente.

Na ausência de filas ele logo ocupa o serviço

O estado do operador muda de livre para ocupado.

O tempo de chegada é guardado em TC (TC= 17,5) e é programado no calendário, um evento de saída, em função do término do serviço.

O tempo de serviço sorteado para o primeiro cliente foi de 11,5 min. Logo a saída é no tempo 29

Calendário de Eventos Futuros

- Assim o evento (1; 29,0; S) indica que o cliente 1 terminará a atividade junto ao operador no tempo 29,0 ocasionando a ocorrência de um evento de saída.
- O tempo sorteado para a próxima chegada é de 7,5 min.
- O calendário de eventos apresenta o evento correspondente à chegada do próximo cliente no tempo programado do relógio 25,0 min.
- O conjunto (2; 25,0; C) no topo do calendário indica que este evento é o evento eminente.

| Num do Client | Tempno relóg | Tipo de Event | EstadFila Client | Estado Operad | TC | CalenEvents Futuros | Cont Clien | SomaTF | Max TF | SomaTS | Max TS |
|---------------|--------------|---------------|------------------|---------------|------|--|------------|--------|--------|--------|--------|
| - | | Início | 0 | Livre | - | (1; 17,5; C) (-;240,0;FS) | 0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1 | 17,5 | Cheg. | 0 | Ocup. (1) | 17,5 | (2; 25,0; C) (1; 29,0; S) (-;240,0;FS) | 0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 2 | 25,0 | Cheg. | 1 (2) | Ocup (1) | 25,0 | (1; 29,0; S) (3; 37,5; C) (-;240,0;FS) | 0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 1 | 29,0 | Saída | 0 | Ocup (2) | - | (3; 37,5; C) (2; 41,6; S) (-;240,0;FS) | 1 | 4,0 | 4,0 | 11,5 | 11,5 |
| 3 | 37,5 | Cheg. | 2 (3) | Ocup (2) | 37,5 | (4;40,0;C) (2; 41,6; S) (-;240,0;FS) | 1 | 4,0 | 4,0 | 11,5 | 11,5 |
| 4 | 40,0 | Cheg. | 2 (3,4) | Ocup (2) | 40,0 | (2; 41,6; S) (5; 42,5; C) (-;240,0;FS) | 1 | 6,5 | 4,0 | 11,5 | 11,5 |
| 2 | 41,6 | Saída | 1 (4) | Ocup (3) | - | (5; 42,5; C) (3;53,6;S) (-;240,0;FS) | 2 | 8,10 | 4,1 | 27,1 | 16,6 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Calendário de Eventos Futuros

- **Calendário de eventos futuros:** chave da simulação. Nele são listados todos os eventos já programados para o futuro
 - (1; 29,0; S) indica que a entidade 1 terminará a atividade junto ao operador no tempo 29,0
 - (2; 25,0; C) indica a chegada do segundo cliente no tempo 25,0
- **Cont. Clientes** – é incrementado sempre que um cliente deixa o sistema
- **Som TF e Soma TS** acumulamos tempos despendidos pelos clientes na fila do servidor e do sistema até ao momento mostrado pela variável do relógio.

Ferramentas voltadas a Simulação (Arena)

PRÓXIMA AULA