

---

---

# Simulação

Transparências de apoio à leccionação de aulas teóricas

**Slide 1**

Versão 1

©2001

José Fernando Oliveira – FEUP

## Sistemas e modelos - porquê simular?

---

- **Sistema** – todo e qualquer objecto sobre o qual se pretende realizar um determinado estudo.
- **Modelo** – representação do objecto na qual se irá, efectivamente executar o estudo.

### Slide 2

Experimentação sobre o próprio sistema, estimulando-o e registando as alterações observadas nos parâmetros que definem o seu estado — maior fiabilidade nos resultados obtidos.

Sistemas demasiado grandes, não acessíveis directamente ou eticamente não experimentáveis — experimentação sobre um modelo: simulação.

*A melhor coisa que se pode fazer, a seguir a observar o sistema real em funcionamento, é fazer uma simulação desse sistema.*

## Exemplos, no “passado”, de aplicações de técnicas de simulação

---

- o exército prussiano simulava guerras através de exercícios de campo onde obrigava os soldados a marchar através das florestas da Europa Central, debaixo de todo o tipo de clima, sob as ordens da cadeia de comando (tradição que ainda hoje todos os exércitos do mundo seguem – as manobras militares);
- simuladores de voo para treino de pilotos de aeronaves (experiências sobre os efeitos da ultrapassagem da barreira do som utilizando dispositivos rotativos onde a aceleração centrífuga substituía a aceleração linear dos jactos);
- no filme *Apollo 13* ilustra-se a utilização de simulação para resolver o problema de determinar qual a melhor sequência de ligação dos dispositivos eléctricos no retorno à atmosfera, dentro das limitações do sistema eléctrico que os astronautas enfrentavam;
- jogos... com dados (Monopólio, Jogo da Bolsa), ou mais recentemente em computador (de guerra, de estratégia, etc.).

### Slide 3

## Aplicações modernas de simulação, tirando partido do poder de cálculo introduzido pelo computador

---

### Slide 4

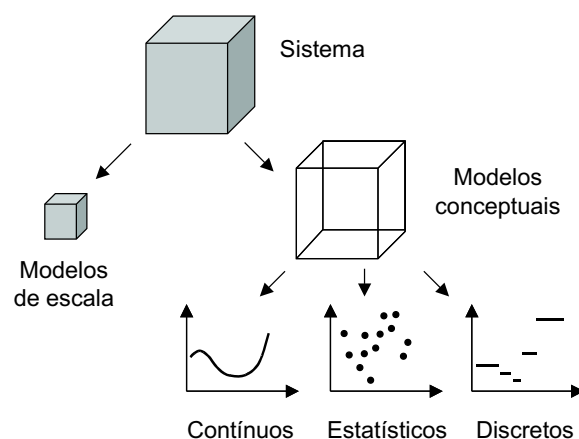
- problemas matemáticos, como: a estimação da área sob uma curva, ou mais genericamente cálculo de integrais múltiplos e a estimação do valor de  $\pi$ ;
- problemas industriais incluindo: projecto de sistemas com filas de espera, redes de comunicação, controlo de stocks, processos químicos;
- problemas económicos e de gestão: comportamento de consumidores, determinação de preços, previsões económicas;
- problemas sociais e comportamentais: dinâmica de populações, efeitos do ambiente na saúde, estudos epidemiológicos, comportamento de grupos;
- sistemas biomédicos: equilíbrio de fluidos, distribuição de electrólitos no corpo humano, proliferação das células sanguíneas, actividades cerebrais;
- estratégias de guerra e táticas militares.

## Tipos de modelos

---

### Slide 5

- **Modelos de escala** – o sistema é substituído por uma versão à escala (uma “maqueta”) do próprio sistema.
- **Modelos conceptuais** – o sistema é substituído por uma versão conceptual do seu funcionamento: um determinado conjunto de equações matemáticas ou relações de causa-efeito.



## Modelos conceptuais

---

Vantajosos face aos modelos de escala porque:

- fisicamente mais transportáveis;
- muito mais versáteis, permitindo mais facilmente alterações na sua estrutura;
- implementáveis com um computador!

**Slide 6** Tipos de modelos conceptuais:

- Determinísticos
  - Contínuos – os estados do sistema variam continuamente com a variável independente.
  - Discretos – o sistema apenas se pode encontrar num número finito de estados, mudando instantaneamente de um estado para outro.
- Estatísticos – o estado do sistema não depende deterministicamente das variáveis de entrada mas a sua resposta é aleatória e caracterizada por um conjunto de distribuições de probabilidade.

## Simulação com modelos contínuos (simulação contínua)

---

Num modelo de simulação contínua o estado  $Y$  de um sistema depende continuamente da variável independente  $x$ , sendo essa relação governada por equações (eventualmente equações diferenciais)  $\Rightarrow$  conhecimento de  $Y = E(x)$ .

*Exemplo: simulação da velocidade de um autocarro, desde que arranca até que pára novamente.*

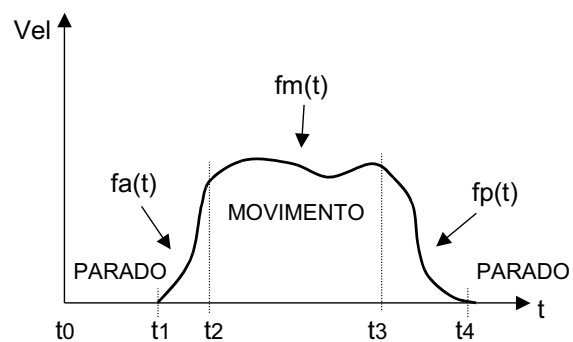
**Slide 7**

Sistema – autocarro

Estado do sistema – velocidade

Variável independente – tempo

$$E(t) = \begin{cases} 0, & t_0 < t < t_1 \\ fa(t), & t_1 \leq t < t_2 \\ fm(t), & t_2 \leq t < t_3 \\ fp(t), & t_3 \leq t < t_4 \end{cases}$$



## Parametrização dos modelos

---

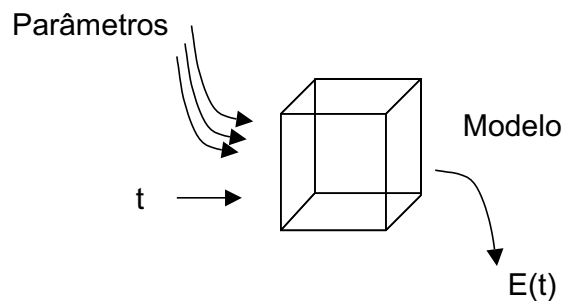
Este modelo apenas nos permite simular a velocidade de um autocarro que arranque e pare naqueles instantes de tempo específicos  $\Rightarrow$  muito redutor, pouco útil.

*Alternativa* – um modelo que tenha como parâmetros de entrada (para além da variável independente  $t$ ):

- o tempo inicial parado:  $dTP$
- o tempo de movimento:  $dTM$
- o tempo de arranque:  $dTA$
- o tempo de paragem:  $dTp$

### Slide 8

$$E(t) = \begin{cases} t_1 = & t_0 + dTP \\ t_2 = & t_1 + dTA \\ t_3 = & t_2 + dTM \\ t_4 = & t_3 + dTp \\ 0, & t_0 < t < t_1 \\ fa(t), & t_1 \leq t < t_2 \\ fm(t), & t_2 \leq t < t_3 \\ fp(t), & t_3 \leq t < t_4 \end{cases}$$



## Simulação com modelos estatísticos (Simulação estatística ou de Monte Carlo)

---

Não se opera sobre um modelo descritivo dos meandros do sistema, mas sim sobre um conjunto de **funções de probabilidade** que caracterizam a forma como o sistema responde.

*Exemplo:* Sistema composto por um dado.

### Slide 9

Simulação determinística – escrita das equações que regem o movimento de um sólido, dados como parâmetros a velocidade inicial, o ângulo com que é atirado, o atrito oferecido pela mesa, etc.

Simulação estatística – amostragem da função de probabilidade de, num dado lance, sair uma determinada face.

## Estimação da área de um círculo

---

Estimar a área do círculo cuja equação é dada por:

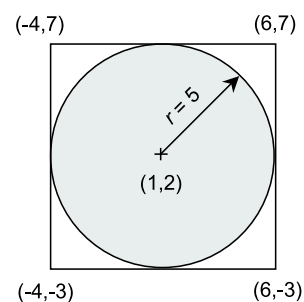
$$(x - 1)^2 + (y - 2)^2 = 25$$

isto é, um círculo de raio 5 e centro em  $(x, y) = (1, 2)$ .

**Procedimento:**

### Slide 10

- encaixar o círculo num quadrado de lado igual ao diâmetro do círculo;
- construir um “gerador” aleatório de pontos do quadrado, que deverão ocorrer com igual probabilidade;
- então, se de uma amostra aleatória de  $n$  pontos,  $m$  calharem dentro do círculo:



$$\text{Área do círculo} \approx \frac{m}{n} \times \text{Área do quadrado} = \frac{m}{n} (10 \times 10)$$

## Estimação da área de um círculo (continuação)

---

Obtêm-se pontos do quadrado igualmente prováveis representando as coordenadas  $x$  e  $y$  de um ponto do quadrado por variáveis aleatórias com as seguintes distribuições:

$$f(x) = \frac{1}{10}, \quad -4 \leq x \leq 6$$

$$g(y) = \frac{1}{10}, \quad -3 \leq y \leq 7$$

### Slide 11

Sejam  $R_1$  e  $R_2$  dois números aleatórios distintos entre 0 e 1. Então um ponto  $(x, y)$  igualmente provável no quadrado será dado por:

$$x = -4 + [6 - (-4)]R_1 = -4 + 10R_1$$

$$y = -3 + [7 - (-3)]R_2 = -3 + 10R_2$$

Finalmente, um ponto aleatório  $(x', y')$  cai dentro do círculo se

$$(x' - 1)^2 + (y' - 2)^2 \leq 25$$

## Números aleatórios

---

### Slide 12

- Um número diz-se aleatório se a sua ocorrência numa sequência de números é imprevisível a partir de quaisquer ocorrências anteriores e se possui a mesma probabilidade de ocorrência ao longo de um intervalo predefinido de valores.
- Fenómenos verdadeiramente aleatórios: ruído radioactivo, ruído de fundo em componentes electrónicos, etc. (sistemas físicos).
- Com computadores apenas se conseguem gerar sequências **pseudo-aleatórias**:

*Utilização de um processo iterativo, realizado sobre uma expressão matemática, que com base no número anterior calcula o número seguinte*

$$n_{i+1} = f(n_i)$$

*de forma a que a relação entre os dois seja o mais imprevisível possível dentro de uma gama grande de valores.*

## Números pseudo-aleatórios

---

$$n_{i+1} = f(n_i)$$

### Slide 13

- Desvantagem – Assim que se repita um número toda a sequência é repetida.
- Vantagem – É possível duplicar exactamente uma sequência de números aleatórios: importante para a detecção e correcção de erros, verificação e validação de um modelo de simulação.

Diferentes funções  $f(n_i)$

↓

Diferentes métodos de geração de números pseudo-aleatórios.

## Geração de números pseudo-aleatórios

---

Método mais comum: **Método congruencial multiplicativo**

$$n_{i+1} = (b \times n_i + c) \bmod(m)$$

em que:

- $mod$  representa o resto da divisão inteira;
- $m$  é o maior número que pode ser gerado (exclusive);
- $b$  e  $c$  são parâmetros da fórmula;
- ao primeiro valor  $n_0$  dá-se o nome de semente do gerador.

Slide 14

Valores concretos para geração de inteiros de 16-bits e 32-bits, respectivamente:

$$f(n_i) = (3993n_i + 1) \bmod(32767) \quad f(n_i) = (16807n_i + 0) \bmod(2147483647)$$

Para gerar um valor entre 0 e 1 basta dividir  $n_{i+1}$  por  $m$ :

$$R_{i+1} = \frac{n_{i+1}}{m}$$

## Tornando o modelo estocástico

---

No problema da determinação da área de um círculo os parâmetros do modelo (centro e raio da circunferência) eram determinísticos, isto é, para um dado ponto era possível responder com certeza se estava dentro ou fora da circunferência.

Parâmetros estocásticos (descritos por funções de probabilidade)

↓

Modelos estocásticos.<sup>a</sup>

↓

Amostragem de funções de probabilidade  $\left\{ \begin{array}{l} \text{histogramas (discretos)} \\ \text{distribuições (contínuas)} \end{array} \right.$

---

<sup>a</sup>Note-se que o modelo continua a ser determinístico, mas porque os seus parâmetros passam a variar estocasticamente ao longo da simulação, a sua resposta deixa de ser determinística e passa a ser estocástica. É como se, no problema da determinação da área do círculo, o raio variasse ao longo da simulação, podendo um mesmo ponto ser declarado fora e dentro do círculo em momentos diferentes da simulação.

Slide 15



## Construção de histogramas a partir de dados experimentais

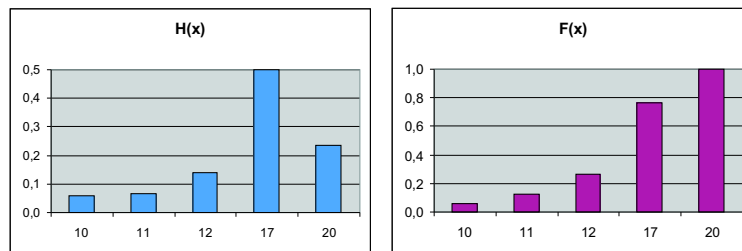
---

Variável observada: número de carros que num minuto entra numa dada rua.  
 Foram realizadas 11 amostras: 17, 20, 12, 17, 17, 10, 17, 20, 11, 12, 17

Classificação dos dados:

Nº de carros (por min)	Total de carros	Percentagem $H(x)$	Percentagem acumulada $F(x)$
10	$1 \times 10 = 10$	5.88	5.88
11	$1 \times 11 = 11$	6.47	12.35
12	$2 \times 12 = 24$	14.12	26.47
17	$5 \times 17 = 85$	50.00	76.47
20	$2 \times 20 = 40$	23.53	100.00

Slide 16

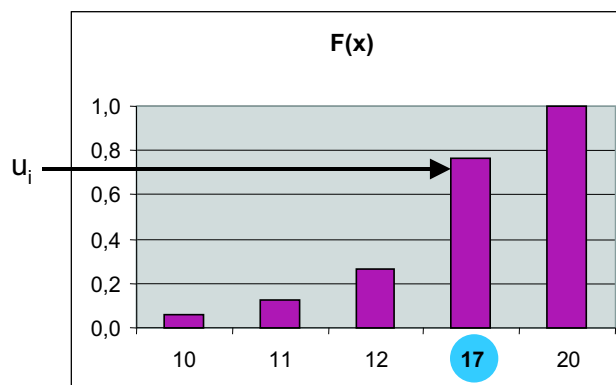


## Amostragem de histogramas

---

A partir de um número aleatório  $u_i$ , pertencente ao intervalo  $[0, 1]$ , obter um dos valores de  $x$  de acordo com as probabilidades induzidas pelo histograma.

Slide 17



Pressuposto: a sequência de números (pseudo-)aleatórios reflecte uma distribuição de probabilidades constante.

## Manutenção de uma linha de produção

---

Uma fábrica possui uma linha de produção que labora 24 horas por dia durante 360 dias por ano. Esta linha de produção gera um valor acrescentado de 500 euros por hora. Por vezes a linha apresenta avarias necessitando de reparação. Históricos detalhados permitiram concluir que o número de horas de funcionamento entre avarias apresenta a seguinte distribuição de probabilidades:

Slide 18

Nº de horas	Probabilidade	Nº de horas	Probabilidade
10	0.05	560	0.05
20	0.05	670	0.05
40	0.05	790	0.05
70	0.05	920	0.05
110	0.05	1060	0.05
160	0.05	1210	0.05
220	0.05	1370	0.05
290	0.05	1540	0.05
370	0.05	1720	0.05
460	0.05	1920	0.05

O tempo de funcionamento médio (entre avarias) é pois de 675.5 horas.

## Manutenção de uma linha de produção (continuação)

---

Após uma avaria o tempo necessário para a reparação é também variável. Se o tempo de reparação ultrapassar os 3 dias então uma unidade de substituição pode ser obtida, ao abrigo do contrato de garantia, para repor a fábrica em funcionamento. Como esta substituição leva um dia, no pior dos casos o tempo de reparação será de 4 dias. A distribuição de probabilidade para o tempo de reparação é dada pela seguinte tabela:

Slide 19

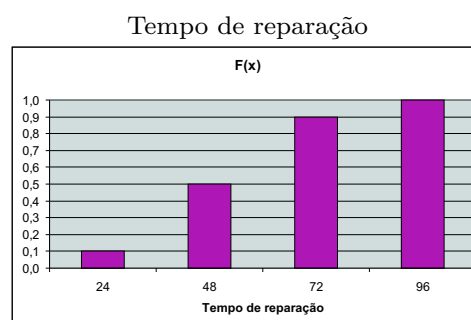
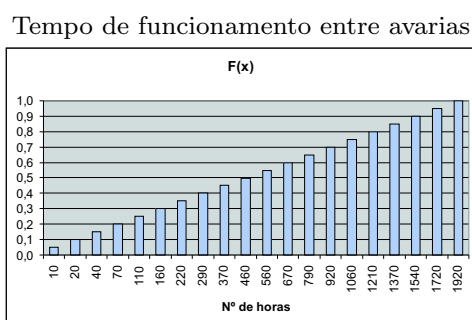
Tempo de reparação (horas)	Probabilidade
24	0.10
48	0.40
72	0.40
96	0.10

O tempo de reparação médio é de 60 horas. Está-se a assumir que o tempo de reparação assumirá sempre um valor múltiplo de 24 horas.

## Manutenção de uma linha de produção – amostragem dos histogramas

Histogramas de probabilidade acumulada

Slide 20



## Manutenção de uma linha de produção – resultados de uma simulação de tamanho 10

Slide 21

Tentativa	Nº de horas em funcionamento	Tempo de reparação (TR)	Tempo total (TT)	% de tempo perdido (TP) $\left(\frac{TR}{TT}\right)$	Avarias $\left(\frac{8640h}{TT}\right)$	Custo anual $(500 \times 8640 \times \% TP)$
1	1720	96	1816	5.29 %	4.758	228 370.04
2	1060	48	1108	4.33 %	7.798	187 148.01
3	1210	72	1282	5.62 %	6.739	242 620.90
4	670	96	766	12.53 %	11.279	541 409.92
5	1720	48	1768	2.71 %	4.887	117 285.07
6	220	24	244	9.84 %	35.410	424 918.03
7	40	48	88	54.55 %	98.182	2 356 363.64
8	920	72	992	7.26 %	8.710	313 548.39
9	110	72	182	39.56 %	47.473	1 709 010.99
10	1210	48	1258	3.82 %	6.868	164 833.07
			Médias	14.55 %	23.210	628 550.81

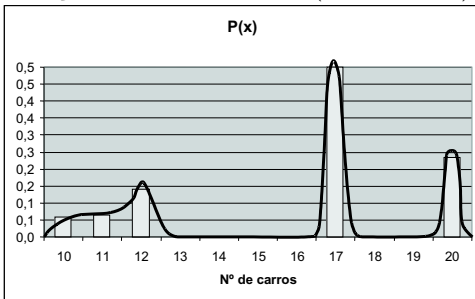
## Amostragem de distribuições (método da inversão)

- Quando a distribuição de probabilidades é dada de uma forma analítica  $P(x)$  e  $x$  varia de uma forma contínua no intervalo dado.
- Origem de  $P(x)$ : ajuste a valores experimentais ou conhece-se o comportamento *a priori* dos parâmetros (estocásticos) do modelo – seguem uma distribuição normal, exponencial negativa, etc.

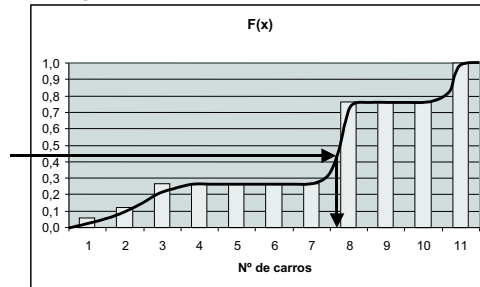
### Slide 22

Ajuste de uma distribuição contínua à distribuição experimental do número de carros:

Função de probabilidade (aproximada)



Função de probabilidade acumulada

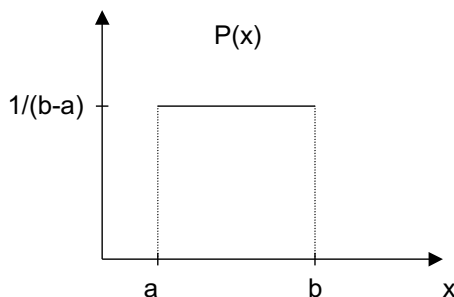


Conhecida a expressão analítica de  $F(x)$ , a determinação do valor de  $x$  para um dado  $u_i \in [0, 1]$  faz-se a partir da inversa de  $F(x)$ :  $x = F^{-1}(u_i)$ .

## Amostragem de distribuições típicas

### 1. Distribuição uniforme

### Slide 23



$$P(x) = \frac{1}{b-a}, \quad a \leq x \leq b$$

$$\begin{aligned} F(x) &= \int_a^x \frac{1}{b-a} dx \\ &= \frac{1}{b-a}(x-a) \\ &\Downarrow \\ F^{-1}(x) &= (b-a)x + a \\ &\Downarrow \end{aligned}$$

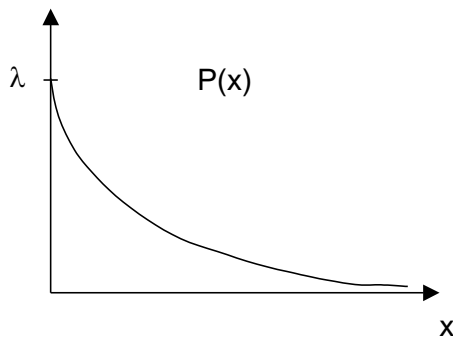
$$x = (b-a)u_i + a$$

(com  $u_i$  uniformemente distribuído sobre o intervalo  $[0,1]$ )

## Amostragem de distribuições típicas (continuação)

### 2. Distribuição exponencial negativa

Slide 24



$$P(x) = \lambda e^{-\lambda x}, x > 0$$

$$\begin{aligned} F(x) &= \int_0^x \lambda e^{-\lambda x} dx \\ &= 1 - e^{-\lambda x} \\ &\Downarrow \\ F^{-1}(x) &= -\frac{\ln(1-x)}{\lambda} \\ &\Downarrow \end{aligned}$$

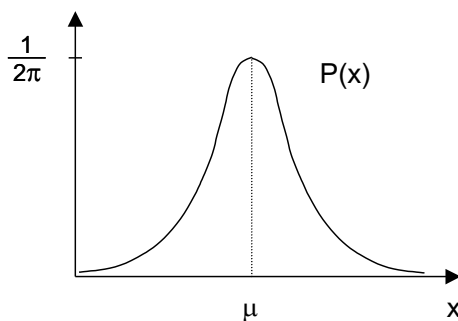
$$x = -\frac{\ln(1-u_i)}{\lambda}$$

(com  $u_i$  uniformemente distribuído sobre o intervalo  $[0,1]$ )

## Amostragem de distribuições típicas (continuação)

### 3. Distribuição normal

Slide 25



$$P(x) = \frac{1}{2\pi} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$F(x) = \frac{1}{2\pi} \int_{-\infty}^x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

não existe forma de cálculo analítica para este integral

↓  
cálculo do integral, e necessariamente de  $F^{-1}(x)$ , por métodos numéricos

↓  
método de Box e Muller

$$x = -2 \ln u_i \cos(2\pi v_i)$$

(com  $u_i$  e  $v_i$  uniformemente distribuídos sobre o intervalo  $[0,1]$  e  $x \rightsquigarrow N(0,1)$ )

Para obter uma variável  $z \rightsquigarrow N(\mu, \sigma^2)$ :

$$z = \mu + \sigma x$$

## Amostragem de distribuições (método da convolução)

**Ideia:** exprimir a amostra pretendida como uma soma estatística (convolução) de amostras de outras distribuições que sejam mais fáceis de amostrar.

Slide 26

## A distribuição de Poisson

$X$  — Número de ocorrências por unidade de tempo  $\Delta t$ .

$$P(X = k) = \frac{e^{-\lambda\Delta t} \cdot (\lambda\Delta t)^k}{k!}$$

com  $k = 0, 1, \dots, n$

- Distribuição **discreta** que descreve a ocorrência de fenómenos aleatórios ao longo do tempo.
- $\Delta t$  terá que verificar algumas condições (ver bibliografia sobre Probabilidades e Estatística).
- $\lambda$  é o número médio de ocorrências por unidade de tempo  $\Delta t$ .

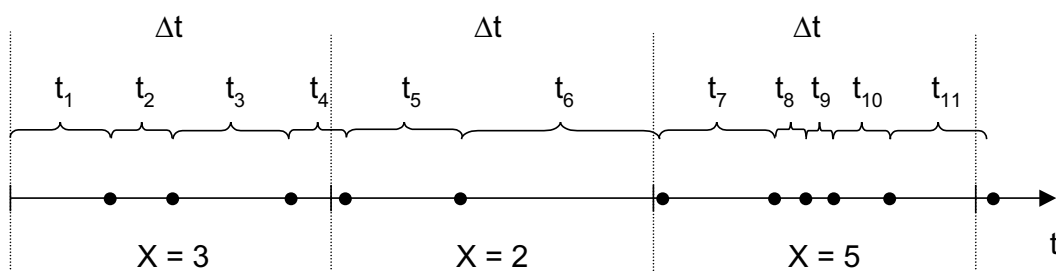
Slide 27

## Amostragem da distribuição de Poisson

É sabido que o tempo  $t$  entre ocorrências sucessivas de um processo de Poisson com parâmetro  $\lambda$  (número de ocorrências por unidade de tempo) segue uma distribuição exponencial negativa também de parâmetro  $\lambda$ .

A partir da distribuição exponencial negativa gera-se uma sequência de  $t_i$  e conta-se quantos cabem nos sucessivos intervalos  $\Delta t$ :

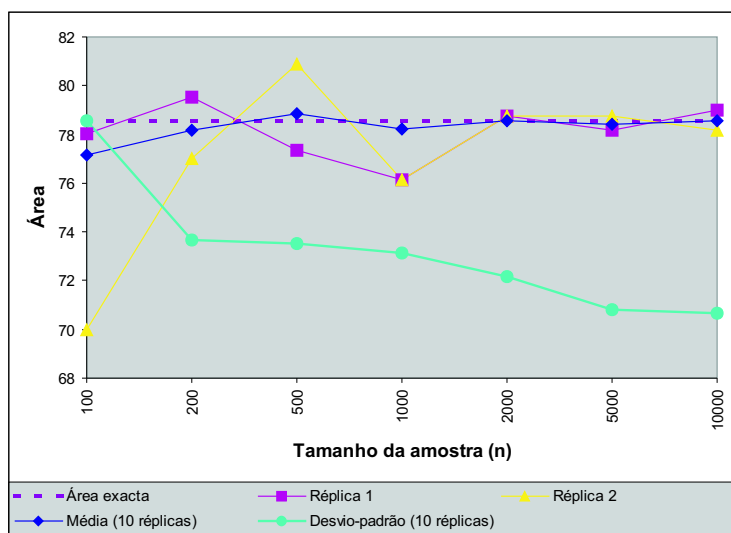
Slide 28



## Obtenção de resultados com um modelo de simulação – caso da estimação da área de um círculo

Quantas vezes se deve repetir a amostragem aleatória para ter uma “boa” estimativa da área do círculo?

Slide 29



Para tamanhos de amostras a variar entre  $n = 100$  e  $n = 10000$ , a experiência foi ainda replicada 10 vezes para cada  $n$ , tendo em cada réplica sido usada uma sequência diferente de números aleatórios.

Resultados de 2 das réplicas e média e desvio-padrão das 10 réplicas.

## Caso da estimação da área de um círculo – conclusões

- A estimativa da área do círculo melhora à medida que o tamanho da amostra aumenta.
- A média das 10 réplicas para cada tamanho de amostra  $n$  fornece uma estimativa melhor do que cada uma das réplicas individualmente.
- A “precisão” do valor médio das 10 réplicas aumenta com o tamanho da amostra  $n$ , como o declínio do desvio-padrão demonstra de uma forma evidente.

Slide 30

## Obtenção de resultados com um modelo de simulação – caso da manutenção de uma linha de produção

Resultados de diferentes simulações (valores médios):

Tamanho da simulação	% de tempo perdido	N <sup>o</sup> avarias por ano	Custo anual
10	14.55%	23.21	628 550.81
500	7.65%	11.12	330 599.00

Slide 31

- Que confiança posso ter nos resultados de uma simulação?
- Que tamanho deve ter a minha amostra (número de tentativas e/ou réplicas) para que os resultados da simulação sejam fiáveis?

Como as simulações apresentam resultados com variações aleatórias é essencial exprimi-los como **intervalos de confiança**.



## Intervalos de confiança

---

- Estimação de parâmetros
  - **Estimador pontual** – uma função dos valores da amostra que dá um valor aproximado para um parâmetro da distribuição da população.

Slide 32

Amostras diferentes darão origem a estimativas diferentes, mesmo usando o mesmo estimador.

Então, qual desses valores coincide com o valor do parâmetro que está a ser estimado? Provavelmente nenhum!

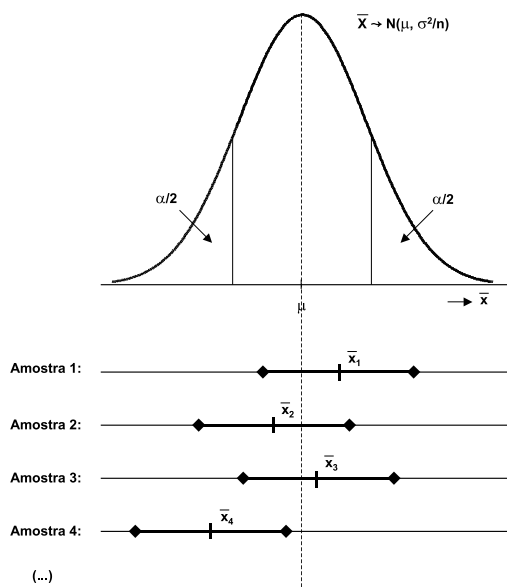
- **Estimação por intervalo**

$$\text{Intervalo de confiança} = \text{estimador pontual} + \text{intervalo} + \text{probabilidade de "acertar"}$$

## Confiança no intervalo

---

Slide 33



Designa-se por  $\alpha$  a percentagem de vezes que, em média, o intervalo não inclui (não “acerta” com) o valor do parâmetro que se pretende estimar.

Para construir um intervalo de confiança é então necessário:

- um estimador do parâmetro em causa
- um valor particular desse estimador
- a distribuição do estimador

## Intervalos de confiança para o valor esperado de uma população

---

O estimador mais natural para o valor esperado de uma população é a média amostral.

### Caso 1 – Amostra grande de uma população qualquer

**Slide 34** No caso de uma amostra grande de uma população com uma distribuição de probabilidade qualquer, com valor esperado  $\mu$  e desvio padrão  $\sigma$ , a média amostral  $\bar{X}$  segue uma distribuição normal, que pode ser convertida numa distribuição reduzida:

$$\bar{X} \rightsquigarrow N\left(\mu, \frac{\sigma^2}{n}\right) \Rightarrow Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \rightsquigarrow N(0, 1)$$

### Amostra grande de uma população qualquer (continuação)

---

Em geral  $\sigma$  é desconhecido mas pode ser substituído, com erro desprezável (por a amostra ser grande), pelo valor dado pelo estimador “desvio padrão amostral”:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$$

**Slide 35**

Logo, assumindo que  $\sigma \approx s$ :

$$Z = \frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \approx \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \rightsquigarrow N(0, 1)$$

Esta distribuição permite-nos calcular o intervalo de confiança para  $\mu$  a  $(1 - \alpha) \times 100\%$ , isto é, um intervalo em torno da média amostral  $\bar{X}$  que, com uma probabilidade de  $(1 - \alpha) \times 100\%$ , contenha o verdadeiro valor esperado da população  $\mu$ .

## Amostra grande de uma população qualquer (conclusão)

Seja então  $z(\frac{\alpha}{2})$  um valor tal que  $P[Z > z(\frac{\alpha}{2})] = \frac{\alpha}{2}$  e seja ainda  $-z(\frac{\alpha}{2})$  o valor simétrico para o qual  $P[Z < -z(\frac{\alpha}{2})] = \frac{\alpha}{2}$ :

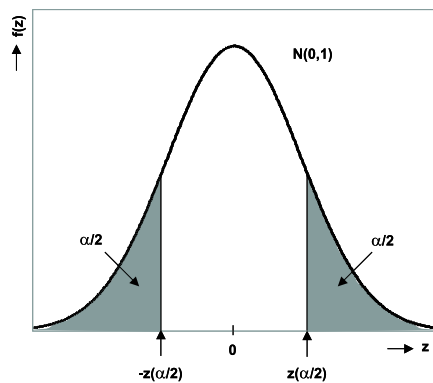
Então:

$$P[-z(\frac{\alpha}{2}) < Z < z(\frac{\alpha}{2})] = 1 - \alpha \quad \Leftrightarrow$$

$$P[-z(\frac{\alpha}{2}) < \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} < z(\frac{\alpha}{2})] = 1 - \alpha \quad \Leftrightarrow$$

$$P[\bar{X} - z(\frac{\alpha}{2}) \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + z(\frac{\alpha}{2}) \frac{\sigma}{\sqrt{n}}] = 1 - \alpha$$

Slide 36



Isto é, o intervalo

$$[\bar{X} - z(\frac{\alpha}{2}) \frac{\sigma}{\sqrt{n}}, \bar{X} + z(\frac{\alpha}{2}) \frac{\sigma}{\sqrt{n}}]$$

é o intervalo de confiança desejado. Considerando que  $s$  aproxima bem  $\sigma$ :

$$[\bar{X} - z(\frac{\alpha}{2}) \frac{s}{\sqrt{n}}, \bar{X} + z(\frac{\alpha}{2}) \frac{s}{\sqrt{n}}]$$

## Caso 2 – Amostra pequena de uma população normal

Quando a amostra é pequena não podemos dizer que o desvio padrão amostral aproxima bem o desvio padrão da população, isto é,  $s \neq \sigma$ . Então  $\frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \not\sim N(0, 1)$ .

Qual é então a distribuição de  $\frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}}$ ?

Slide 37

Demonstra-se que, no caso em que a amostra é pequena mas a população segue uma distribuição normal a média amostral reduzida segue uma distribuição de t de Student com  $n - 1$  graus de liberdade:

$$\frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \sim \frac{N(0, 1)}{\sqrt{\frac{\chi_{n-1}^2}{n-1}}} \sim t_{n-1}$$

Os intervalos de confiança para  $\mu$  a  $(1 - \alpha) \times 100\%$  serão então dados por:

$$\left[ \bar{X} - t_{n-1}(\frac{\alpha}{2}) \frac{s}{\sqrt{n}}, \bar{X} + t_{n-1}(\frac{\alpha}{2}) \frac{s}{\sqrt{n}} \right]$$

## Intervalos de confiança para a variância de uma população

---

Intervalos de confiança para a variância de uma população, em particular quando segue uma distribuição normal, poderiam também ser deduzidos. No entanto, dados os conceitos avançados de Estatística que envolvem, saem do âmbito desta breve introdução à determinação de intervalos de confiança para efeitos de análise de resultados de simulações.

Slide 38

## Dimensionamento das amostras

---

Associado a um intervalo de confiança estão sempre associados dois parâmetros, que dependem da dimensão da amostra:

- a amplitude do próprio intervalo de confiança, que, para um mesmo grau de confiança, será tanto menor quanto maior for a amostra;
- o grau de confiança do intervalo que, para uma mesma amplitude, será tanto maior quanto maior for a amostra.

Slide 39

Em aplicações de simulação o caso mais frequente é o de pretendermos **dimensionar a amostra** de forma a obtermos um intervalo de confiança com um **grau de confiança fixado** e dado à partida e uma **amplitude do intervalo não superior a** um outro valor.

## Dimensionamento das amostras (continuação)

---

Tomemos como exemplo um intervalo de confiança para o valor esperado da população:

$$\left[ \bar{X} - z\left(\frac{\alpha}{2}\right) \frac{s}{\sqrt{n}}, \bar{X} + z\left(\frac{\alpha}{2}\right) \frac{s}{\sqrt{n}} \right]$$

### Slide 40

Suponhamos que pretendemos um intervalo de confiança a 95%. Neste caso  $\alpha/2 = 0.025$  e tirar-se-ia da tabela da distribuição normal reduzida o valor de  $z$  respectivo:  $z(0.025) = 1.96$ .

**Problema:** queremos um valor  $n$  para saber quantas repetições da simulação se devem fazer (o tamanho da amostra), mas para o determinar precisamos de um valor de  $\bar{X}$  e  $s$  (i.e. os resultados de uma amostra).

**Solução:** retirar uma primeira amostra com uma dimensão qualquer, calcular os parâmetros  $\bar{X}$  e  $s$  e utilizar esses parâmetros para calcular o tamanho da amostra para a experiência seguinte.

## Dimensionamento das amostras (conclusão)

---

**Exemplo:** de uma primeira amostra obtivemos  $\bar{X} = 1.70$  e  $s = 0.051$ . O intervalo de confiança fica então com o seguinte aspecto:

$$\left[ 1.70 - 1.96 \frac{0.051}{\sqrt{n}}, 1.70 + 1.96 \frac{0.051}{\sqrt{n}} \right]$$

### Slide 41

Finalmente, impondo uma amplitude máxima para o intervalo, digamos 0.04, poderíamos calcular o tamanho da amostra necessária:

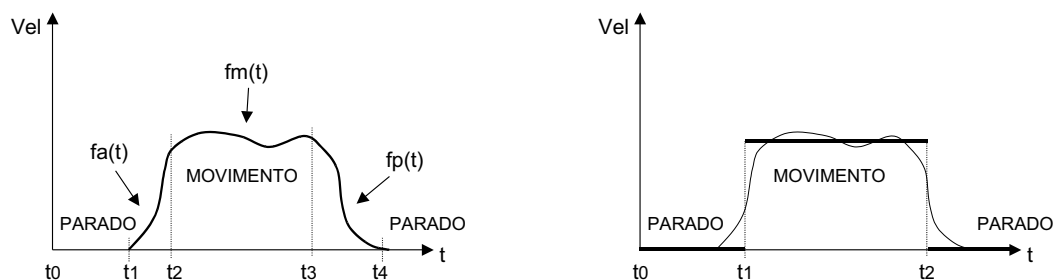
$$\begin{aligned} 1.70 + 1.96 \frac{0.051}{\sqrt{n}} - 1.70 + 1.96 \frac{0.051}{\sqrt{n}} &\leq 0.04 \\ 3.92 \frac{0.051}{\sqrt{n}} &\leq 0.04 \\ n &\geq 25 \end{aligned}$$

## Simulação com modelos discretos ou Simulação discreta

Simulação discreta – aplica-se a sistemas cujos estados variem de forma discreta ao longo dos valores da variável independente:

- por serem intrinsecamente discretos
- por serem uma “discretização” de um sistema contínuo

Slide 42



## Elementos da simulação discreta

Em simulação discreta um sistema descreve-se através de:

- **Entidades**
- que possuem **atributos**
- e levam a cabo **actividades** no sistema.
- Essas actividades, ao começarem ou terminarem, dão origem a **eventos**.
- As entidades, enquanto esperam as condições para levarem a cabo as suas actividades, esperam em **filas de entidades** ou **filas de espera**.

Slide 43

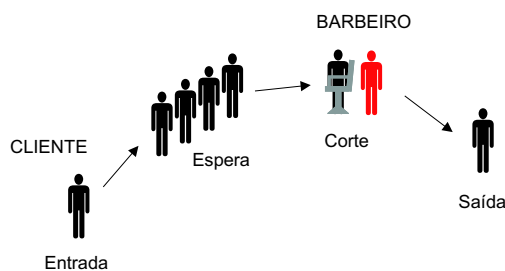
## Entidades

---

**Entidade** – Qualquer elemento capaz de contribuir para o conjunto total de estados do sistema.

Conjunto de estados do sistema = Conjunto de estados das entidades que fazem parte do sistema

### Slide 44



		Entidades	
		BARBEIRO	CLIENTE
Estados	BARB_CORTE	CLIENT_ENTRADA	
	BARB_LIVRE	CLIENT_ESPERA	
		CLIENT_CORTE	
		CLIENT_SAIDA	

## Classificação das entidades

---

- quanto às operações que podem realizar:
  - activas – são responsáveis por gerar transições nos estados do sistema
  - passivas – só alteram o seu estado por intermédio de acções de outras entidades

### Slide 45

- quanto à duração das sua permanência no sistema:
  - permanentes – permanecem no sistema durante todo o processo de simulação
  - temporárias – entram no sistema, são processadas e, provavelmente, saem mais tarde

## Atributos das entidades

---

Valores ou propriedades características da entidade.

Exemplo: Os atributos do barbeiro poderiam ser o tempo médio, e o respectivo desvio-padrão, do tempo que demora a cortar o cabelo a um cliente.

Slide 46

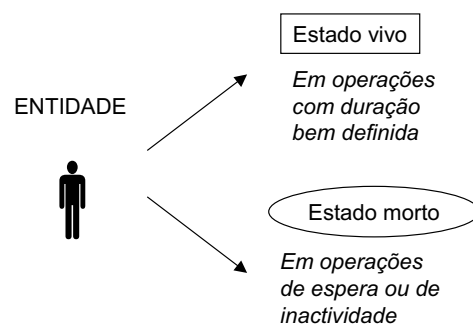
## Actividades

---

São as operações ou procedimentos iniciados em cada instante relevante do sistema, sendo através delas que se transformam os estados das entidades.

### Classificação das actividades:

- Actividade viva – actividade que coloca o sistema num estado vivo
- Actividade morta – actividade que coloca o sistema num estado morto ou de espera



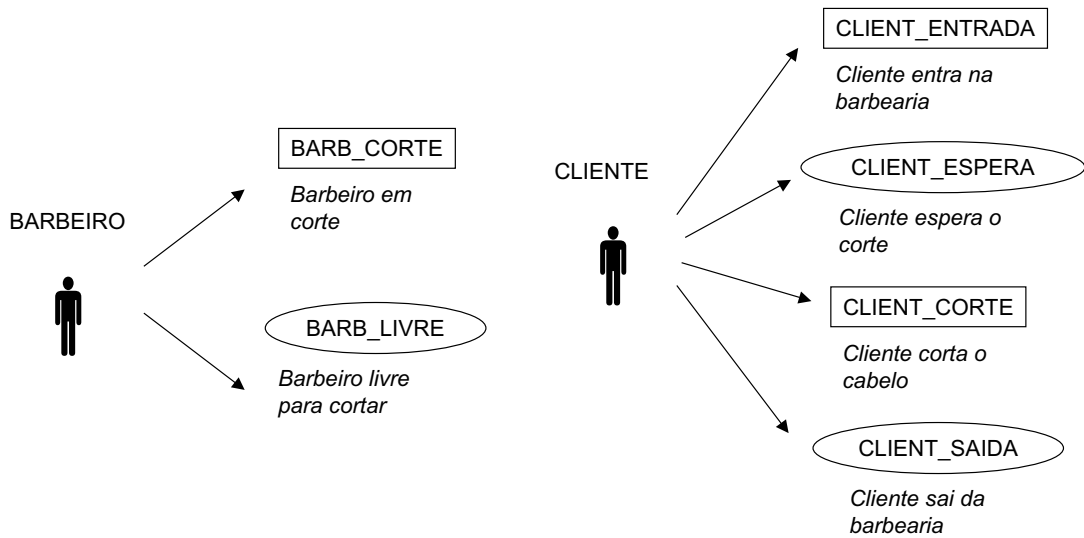
Slide 47

Uma **actividade** corresponde a um **estado dinâmico** (que varia ao longo da simulação) de uma ou várias entidades.



## As actividades do sistema barbearia

Slide 48

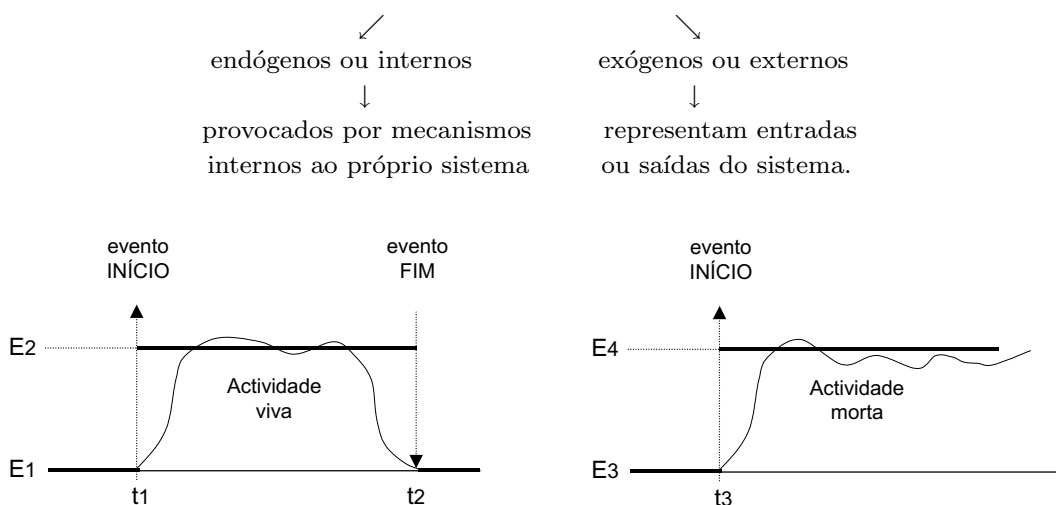


- 1) O sistema evolui ao longo das actividades das suas entidades.
- 2) As entidades interagem umas com as outras através de actividades comuns (e.g. actividade CORTE das entidades BARBEIRO e CLIENTE)

## Eventos

**Evento** – transição do sistema, i.e. algo que acontece num dado instante de tempo e que, através de uma acção instantânea associada ao evento, provoca uma mudança de estado no sistema.

Slide 49



Actividades e eventos são duas perspectivas diferentes mas equivalentes de olhar para a dinâmica de um sistema discreto.

## Filas de entidades ou de espera

---

Locais onde as entidades são colocadas enquanto aguardam (ou esperam) pelo início de uma actividade viva

*isto é*

são a forma típica de modelar os estados mortos de entidades.

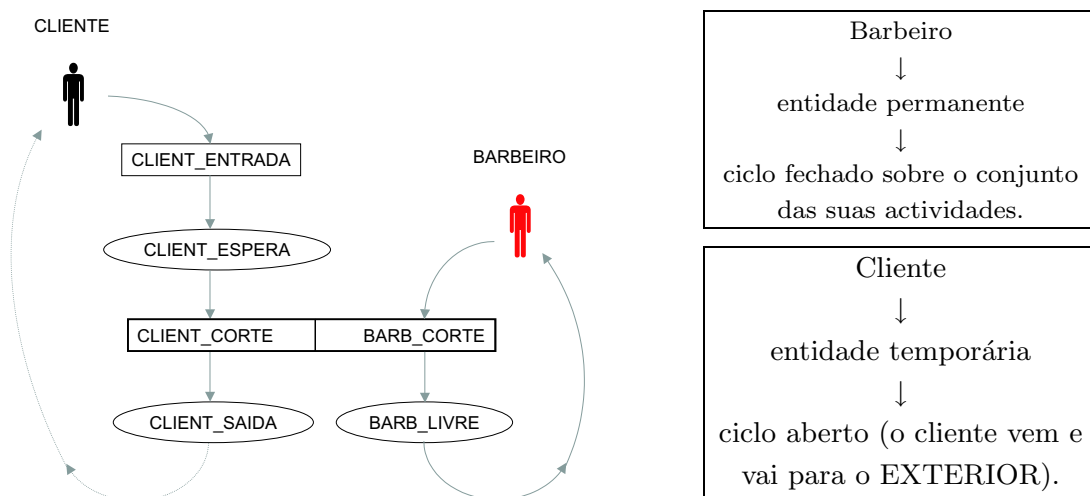
**Slide 50** Filas de espera – tabelas onde as entidades são colocadas por ordem de chegada e retiradas segundo um critério pré-definido (*first-in-first-out*, *last-in-first-out*, etc.)

## Ciclos de actividades

---

Encadeamento com que o conjunto de actividades elementares de uma entidade é executado.

**Slide 51**



Considerando o exterior como um estado (equivalente a CLIENT\_SAIDA) a entidade CLIENTE passa a ter um ciclo fechado de actividades.

As entidades CLIENTE e BARBEIRO possuem uma actividade comum.

## Ciclos de actividades

---

Entre duas actividades *vivas* deve considerar-se sempre um estado de *espera* (actividade *morta*), podendo, no entanto, a entidade passar por essa *espera* directamente para a actividade *viva* seguinte.

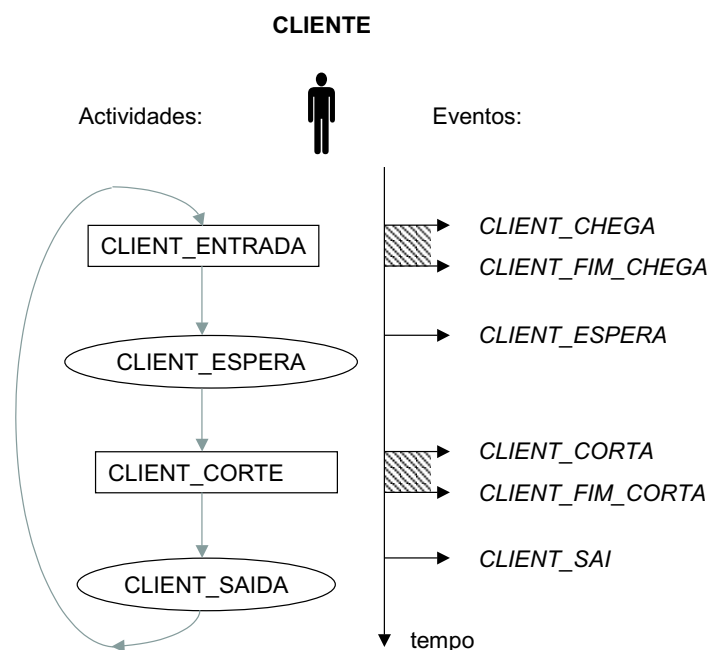
### Slide 52

A modelação usando ciclos de actividades é principalmente usada para sistemas com uma forte componente de filas de espera, como os sistemas de produção, de logística ou de distribuição.

## Relação do ciclo de actividades com o encadeamento de eventos

---

### Slide 53



## A relação do ciclo de actividades com o encadeamento de eventos no sistema barbearia – algumas notas

---

- À actividade viva *CLIENT\_ENTRADA* corresponde uma duração bem determinada  $\Rightarrow$  quando acontece o evento *CLIENT\_CHEGA* pode-se desde logo calcular o instante em que acontecerá o evento *CLIENT\_FIM\_CHEGA* e marcá-lo numa escala de tempo para acontecer no futuro.

### Slide 54

- A passagem da actividade de entrada para a actividade de espera é automática  $\Rightarrow$  um dos eventos *CLIENT\_FIM\_CHEGA* ou *CLIENT\_ESPERA* é redundante, logo pode ser eliminado do modelo.
- O evento *CLIENT\_ESPERA* dá início a um estado morto. Como calcular o instante em que se dará início à actividade viva seguinte (corte)?  
Antes de um *cliente* se colocar na espera, se não houver ninguém à espera e o *barbeiro* estiver livre, marca o evento de início do seu próprio corte. Se não coloca-se em espera e deixa a cargo do *barbeiro* chamá-lo para o corte. Para tal o *barbeiro* terá que examinar a fila sempre que termina uma operação de corte.
- O evento *CLIENTE\_SAI* é coincidente com o evento *CLIENT\_FIM\_CORTE*.

## Técnicas de avanço do tempo

---

~~Avanço no tempo~~  $\Rightarrow$  avanço da variável independente da simulação



Dinâmica do sistema.

Evoluir do sistema ao longo da variável independente.

A maior parte das vezes a variável independente é o tempo



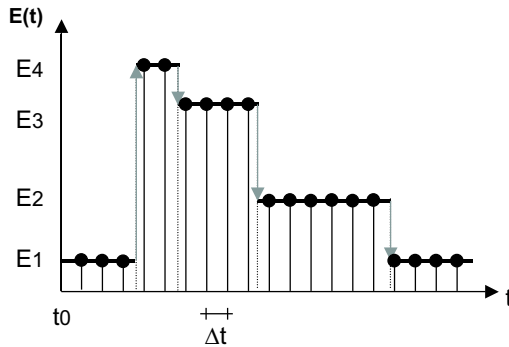
**Avanço no tempo**

### Slide 55

## Avanço regular (*Time slicing*)

Inspecciona o estado do sistema através de amostragens sucessivas em intervalos de tempos regulares.

Slide 56



**Vantagem:** Simplicidade de implementação.

**Limitação principal:** não há sincronismo entre as transições de estado do sistema e o processo de amostragem.

**Solução:** aumentar a frequência de amostragem.<sup>a</sup>

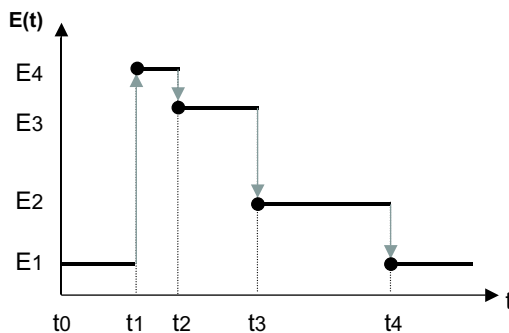
**Problema:** fraca eficiência – pode-se estar ao longo de muito tempo a recolher amostras do mesmo estado... número grande de amostras supérfluas, sem informação.

<sup>a</sup>A lei de Nyquist estabelece o número mínimo de amostras que se deve fazer a um fenómeno para que ele possa posteriormente ser reconstruído: 2 vezes a maior frequência envolvida no sinal que representa o fenómeno.

## Avanço para o próximo evento (*Next event*)

O tempo de simulação avança para o próximo evento.

Slide 57



**Vantagem:** independente das frequências envolvidas nas mudanças de estado, reduzindo ao mínimo o número de amostras recolhidas na simulação.

**Desvantagem principal:** complexidade de implementação – cada vez que se dá um evento é necessário calcular o tempo em que surgirá o próximo.

⇒ Mecanismos de marcação e memorização de eventos futuros.

## Abordagens à criação e concepção de um modelo de simulação discreta

---

Diferentes formas de implementar os diagramas de estados do sistema



Diferentes abordagens à criação e concepção do modelo

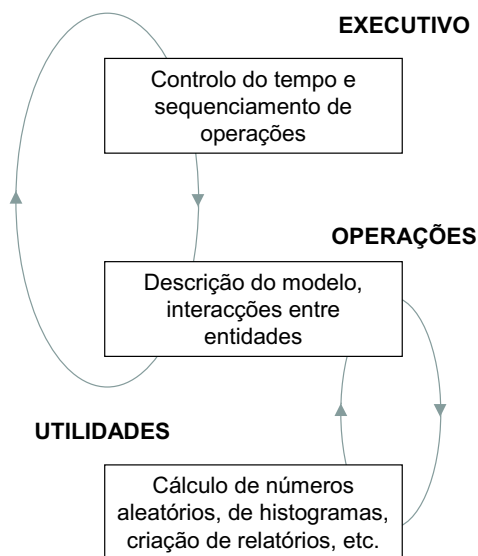
### Slide 58

- Abordagem por actividades
- Abordagem por eventos
- Abordagem por processos
- Abordagem das 3 fases (combinação de actividades e eventos)

## Estrutura de um simulador discreto

---

### Slide 59



Blocos das *operações* e das *utilidades* – conjuntos de rotinas

Bloco *executivo* – ciclo de programa responsável por mandar executar as rotinas do bloco das *operações* – marca a cadência da simulação.

Nas ferramentas comerciais de simulação nem sempre todos os blocos estão acessíveis ao utilizador.

## Abordagem por actividades

---

Criar um modelo:

- Descrever o conjunto de actividades detectadas no sistema...
- ... especificando detalhada e pormenorizadamente a sequência de acções a realizar em cada actividade.
- Só se descrevem as actividades vivas. As actividades mortas são entendidas como filas de espera.

Slide 60

## O bloco das operações – rotinas das actividades

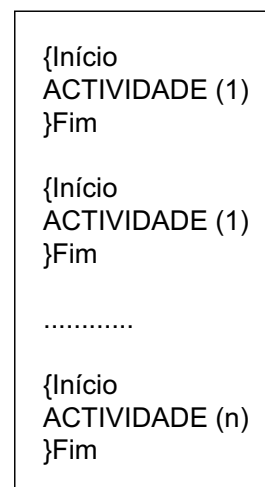
---

- Uma actividade  $\Leftrightarrow$  bloco de programa/rotina independente dos outros.
- Conjunto das rotinas = bloco das operações
- Executivo  $\rightarrow$  percorre o conjunto de rotinas, fazendo executar as que, nesse momento, reúnam condições para tal.

$\Downarrow$

As rotinas devem permitir o teste das suas condições de exequibilidade.

EXECUTIVO  
 $\uparrow \downarrow$   
 BLOCO DAS OPERAÇÕES



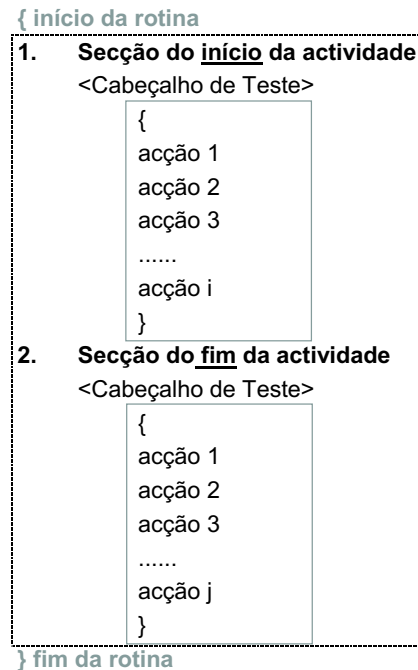
Slide 61

## Decomposição das rotinas de actividades...

... em 2 secções com duas partes distintas.

### Slide 62

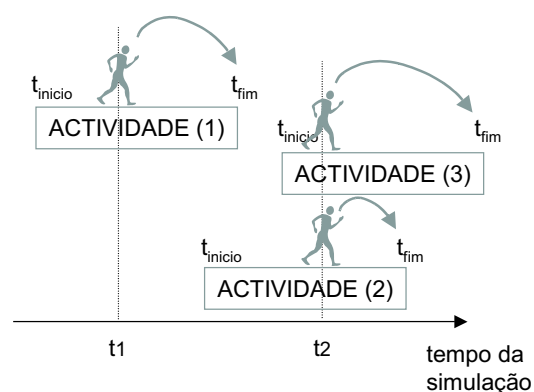
O executivo percorrerá por ordem cada uma das rotinas, tentando executar os procedimentos de início de actividade ou de fim de actividade, comparando as condições actuais da simulação com as condições pertencentes aos cabeçalhos de teste.



## Células de tempo

### Slide 63

- Variável, pertencente ao grupo de atributos de cada entidade, usada como marcador de tempo ( $t_{fim}$ ).
- É através dela que o tempo de permanência dessa entidade numa dada actividade é controlado.
- Cronómetro interno da entidade que assinala o tempo em que deve terminar a actividade em que está envolvida.



Os corpos das rotinas de quaisquer actividades só serão executados quando o tempo de simulação coincidir com o início ( $t_{início}$ ) ou com o fim ( $t_{fim}$ ) de pelo menos uma das actividades do sistema.

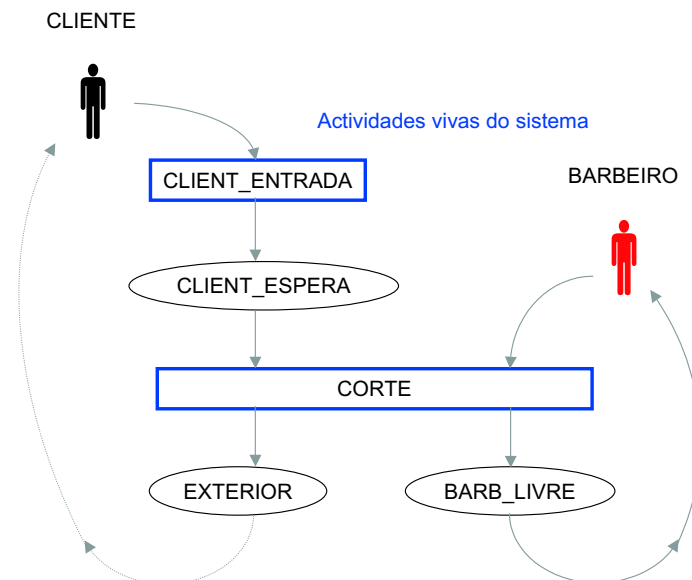


## Abordagem por actividades – o caso da barbearia

---

Ciclo de actividades do cliente e do barbeiro, interagindo na actividade comum CORTE:

Slide 64



As actividades mortas serão modeladas como filas de espera.

Slide 65

## Rotina da actividade de entrada do cliente na barbearia

Slide 66

```

Cliente_entrada()
{//início
// primeiro teste: actividade pode iniciar agora?
if (EntActual.estado==Exterior && EntActual.tfim==tsim)
    {
    1. retira cliente actual do EXTERIOR
    2. calcula o seu tempo de entrada (dT)
    3. cliente.tfim = tsim+dT //marca o fim da sua entrada
    4. coloca-o na fila CLIENT_ENTRADA
    5. cria próximo cliente (cliente2)
    6. calcula o tempo em que ele chegará (tnext)
    7. cliente2.tfim = tnext //marca-o para entrar no futuro
    8. coloca-o no EXTERIOR
    }
// segundo teste: actividade pode finalizar agora?
if (EntActual.estado==CLIENT_ENTRADA && EntActual.tfim==tsim)
    {
    1. passa cliente de CLIENT_ENTRADA para CLIENT_ESPERA
    }
}
}

```

## Rotina da actividade de entrada do cliente na barbearia

Slide 67

- Tudo começa com a colocação de um cliente no EXTERIOR. A partir daí o ciclo do cliente é iniciado e arranca todo o processo de simulação.
- É da responsabilidade da próxima actividade viva retirar o cliente da fila CLIENT\_ESPERA para dar início à operação seguinte (se e quando tal for possível).
- Esta actividade *cliente\_entrada()* só tem interesse nesta forma se o cliente demorar um tempo significativo entre a entrada na barbearia e a chegada à fila de espera para o atendimento.
- No entanto, mesmo que tal não aconteça ( $dT=0$ ), a rotina funciona correctamente sendo a secção de *fim* executada automaticamente a seguir à secção de *início*.
- Nesse caso ( $dT=0$ ) a rotina pode ser reescrita do seguinte modo, apenas com uma secção de início:

## Rotina da actividade de entrada do cliente na barbearia, com $dT=0$

---

Slide 68

```
Cliente_entrada()
{//início
// primeiro teste: actividade pode iniciar agora?
if (EntActual.estado==Exterior && EntActual.tfim==tsim)
{
1. retira cliente actual do EXTERIOR
2. coloca-o na fila CLIENT_ESPERA //directo à espera
3. cria próximo cliente (cliente2)
4. calcula o tempo em que ele chegará (tnext)
5. cliente2.tfim = tnext //marca-o para entrar no futuro
6. coloca-o no EXTERIOR
}
};//fim
```

## Rotina da actividade de entrada do cliente na barbearia

---

Tarefas que qualquer rotina de actividades de entrada devem cumprir:

- Criar a próxima entidade a entrar.  
(seja com base em dados tabelados, na amostragem de histogramas ou por intermédio de valores retirados de qualquer distribuição estatística).

Slide 69

- Assegurar o interface com a próxima actividade morta.

## Rotina da actividade de corte na barbearia

---

Slide 70

```

Corte()
{//início
// Cabeçalho de teste de início da actividade:
// sempre que haja algum cliente à espera e barbeiro livre
if (#CLIENT_ESPERA>0 && barbeiro==BARB_LIVRE)
{
1. retira um cliente da fila CLIENT_ESPERA
2. coloca-o no estado CLIENT_CORTE
3. retira barbeiro de BARB_LIVRE
4. coloca-o no estado BARB_CORTE
5. calcula o tempo em do corte (tcorte)
6. barbeiro.tfim = tsim+tcorte //marca o fim do corte
}
// Cabeçalho de teste de fim da actividade:
// se existe um barbeiro em corte e se o fim de corte
// coincide com o tempo actual da simulação
if (EntActual.estado==BARB_CORTE && EntActual.tfim==tsim)
{
1. retira o cliente do estado CLIENT_CORTE
2. e elimina-o (sai da barbearia)
3. retira barbeiro de BARB_CORTE
4. e coloca-o no estado BARB_LIVRE
}
};//fim

```

## Rotina da actividade de corte na barbearia

---

Slide 71

- Normalmente as actividades vivas são separadas em duas: uma que corresponde ao *início* e outra que corresponde ao *fim* da actividade, o que para o caso da barbearia isto originaria 3 actividades em vez de 2.
- Seguindo esta abordagem, cada rotina tem um só cabeçalho de teste e um corpo de procedimentos.
- A vantagem principal é permitir executar primeiro as acções de finalização das actividades, libertando mais cedo as entidades nelas envolvidas.

## Abordagem por actividades – Recolha de dados

---

Cria-se uma nova actividade (e logo uma entidade), independente das outras, e faz-se executá-la com a frequência que nos interessa recolher os dados.

No caso da barbearia, supondo que nos interessa saber as quantidades de entidades que, num dado momento, se encontram em cada estado:

Slide 72

```
Amostragem()
{//início
if (EntActual==Amostrador &&
EntActual.tfim==tsim)
{
1. memoriza número de entidades no EXTERIOR
2. memoriza número de entidades em CLIENT_ESPERA
3. memoriza número de entidades no CLIENT_CORTE
4. memoriza número de entidades no BARB_CORTE
5. calcula o tempo da próxima amostragem (tnext)
6. amostrador.tfim = tnext //marca próxima amostragem
}
};//fim
```

## Abordagem por actividades – o executivo

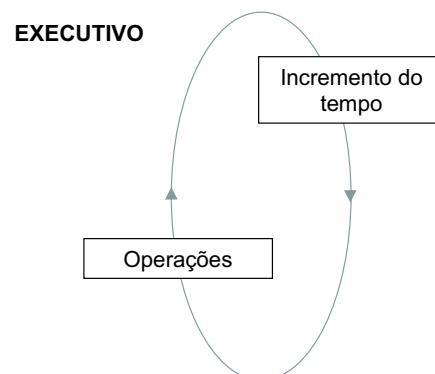
---

- O executivo mantém um constante registo das entidades no sistema – lista de entidades.
- Cada entidade tem uma célula de tempo, onde se encontra registado o tempo em que essa entidade irá sofrer a próxima mudança de estado.

Slide 73



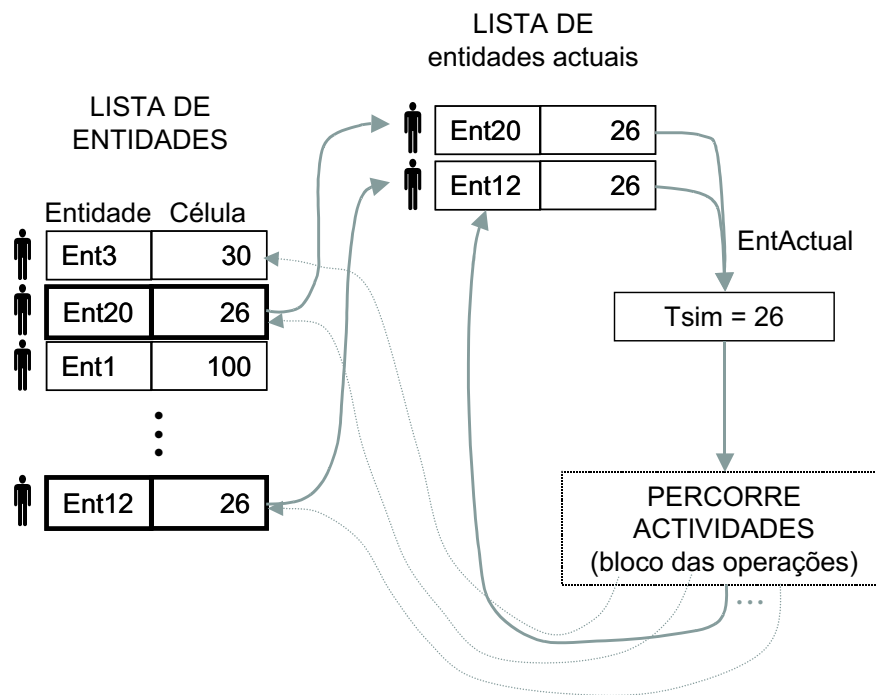
Incremento do tempo: detectar a entidade que apresenta uma célula de tempo com menor valor e colocar o tempo de simulação igual a esse valor → próxima transição do sistema.



## Abordagem por actividades – o executivo

---

Slide 74



## Abordagem por actividades – o executivo

---

Slide 75

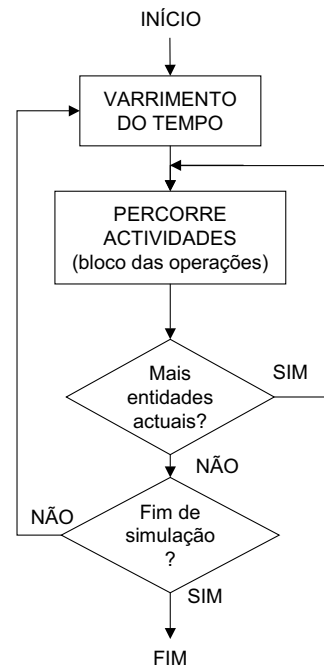
- O executivo percorre a lista de entidades e detectou que a menor célula de tempo era 26. O tempo de simulação é colocado em 26.
- Colocou as entidades *Ent20* e *Ent12* na Lista de Entidades Actuais e percorre o bloco das operações 2 vezes: uma com  $EntActual = Ent20$  e outra com  $EntActual = Ent12$ .
- Durante a execução das actividades tanto as células de tempo como os estados das entidades poderão ser alterados.
- No fim o executivo torna a inspeccionar a lista de entidades procurando o menor valor de célula de tempo.

## Esquema geral do executivo na abordagem por actividades

Slide 76

```

Executivo()
{ //Início
while (simular==TRUE)
  { //VARRIMENTO DO TEMPO:
    {
      1. percorre lista de entidades no sistema
      e detecta aquelas com célula de tempo
      com valor mínimo (tmínimo)
      2. coloca essas entidades na Lista de
      Entidades Actuais
      3. coloca tempo de simulação
      tdim = tmínimo
    }
    while (#ListaEntidadesActuais > 0)
    //enquanto houver EntActual
    {
      1. EntActual = próxima entidade da lista
      //VARRIMENTO DAS ACTIVIDADES
      2. BlocoOperacoes(EntActual, tsim);
      //executa bloco das operações passando-lhe
      //a entidade actual e o tempo da simulação
    }
  }
} //fim
    
```



## Algumas notas finais sobre a abordagem por actividades

Slide 77

- A ordem pela qual as rotinas do bloco das operações são percorridas/analizadas pelo executivo é relevante: as prioridades devem ser estabelecidas pelo analista.
- As secções de *fim* devem ser executadas antes das secções de *início*, para que as entidades nelas envolvidas sejam mais rapidamente libertadas e se aumente a eficiência do processo de simulação.
- A menor eficiência desta abordagem reside no facto de todas as actividades serem analisadas em cada incremento do tempo de simulação, mesmo aquelas para as quais havia à partida informação suficiente para se saber que nem sequer precisavam de ser analisadas.
- Vantagem principal: permite uma implementação simples e rápida das actividades do sistema.

## Abordagem por eventos

---

Criar um modelo:

Slide 78

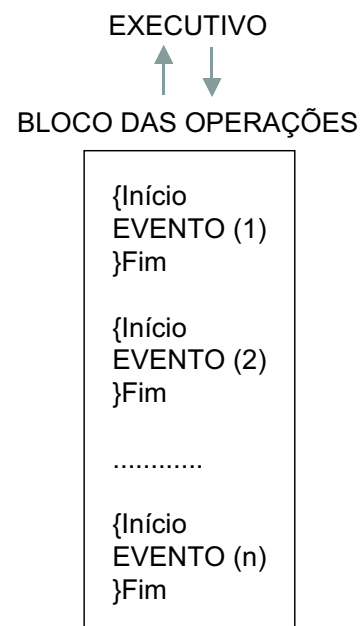
- Descrever os eventos (associados às transições entre estados) relevantes para a representação da dinâmica do sistema...
- ... e o conjunto de acções “instantâneas” que é associado a cada um deles, ...
- ... acções essas que serão executadas sempre que esse evento surja na simulação...
- e através das quais, nomeadamente, outros eventos são marcados para acontecer no futuro.

## O bloco das operações – rotinas dos eventos

---

Slide 79

- Um evento  $\Leftrightarrow$  bloco de programa/rotina independente dos outros, utilizada pelo executivo para fazer executar as acções associadas a esse evento.
- O executivo **não** percorre todas as rotinas em cada incremento de tempo. As rotinas de eventos não necessitam de obedecer a qualquer estrutura rígida de organização interna pois em cada ciclo de simulação apenas uma rotina é executada.

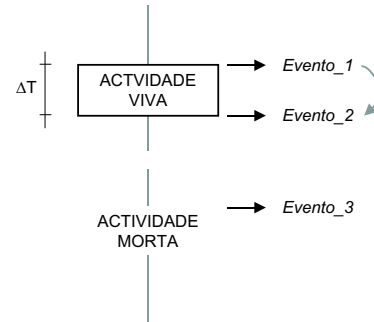


As rotinas dos eventos não necessitam de incluir cabeçalhos de teste pois, a partir do momento em que ele é marcado na escala do tempo de simulação, ele deverá ser sempre executado.



## Relação eventos – actividades

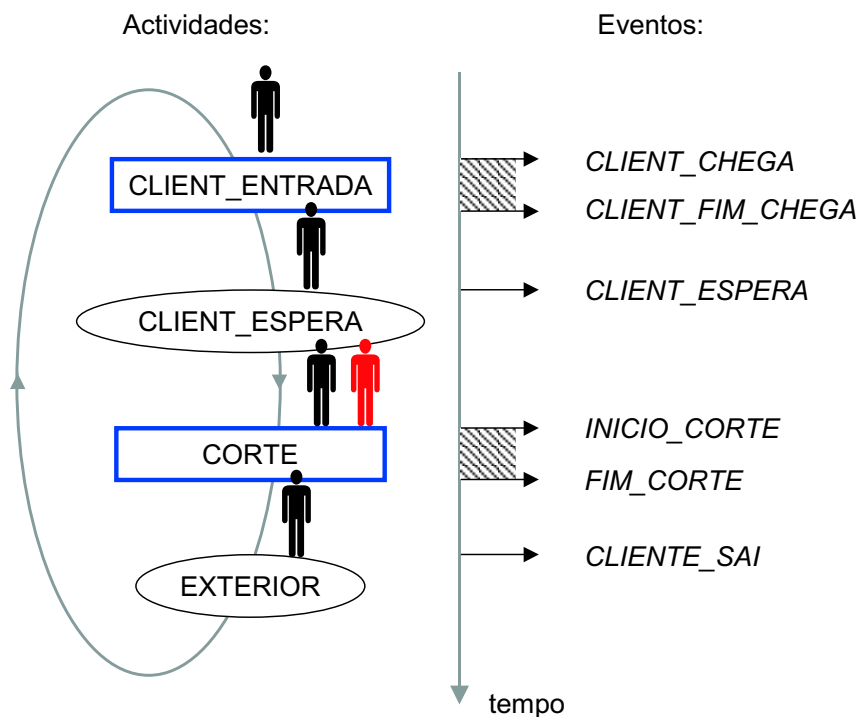
- Das rotinas de *início* de actividades vivas não só se espera uma mudança de estado para alguma entidade, como também a marcação do evento de *fim* dessa mesma actividade.



Slide 80

- Quando o evento corresponde ao *início* de uma actividade morta não é forçoso que se marquem eventos no futuro.

## Abordagem por eventos – o caso da barbearia

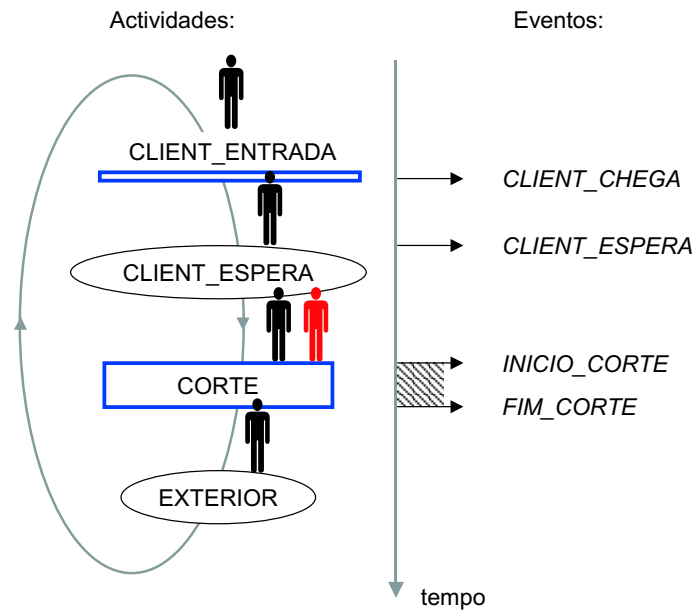


Slide 81

## Abordagem por eventos – o caso da barbearia

Eliminado os eventos irrelevantes, por não se perder tempo entre o fim de uma actividade viva e o início da actividade morta imediatamente seguinte e por se assumir que a entrada é uma actividade instantânea:

Slide 82



## Rotina do evento de chegada do cliente à barbearia

Slide 83

```

Cliente_chega()
{//início
  1. retira cliente actual do EXTERIOR
  2. coloca-o na fila CLIENT_ESPERA //directo à espera
  3. cria próximo cliente (cliente2)
  4. calcula o tempo em que ele chegará (tnext)
  5. Marca(Cliente_chega, tnext) //marca-o para chegar no futuro
  6. coloca-o na fila de EXTERIOR
  //Se só existir o cliente actual na espera e o barbeiro se
  //encontrar livre, então poderá marcar para o tempo actual da
  //simulação o evento de início de corte:
  7. if(#CLIENT_ESPERA==1 && barbeiro==BARB_LIVRE)
    {
      Marca(Inicio_corte, tsim) //marca já o início do corte
    }
}
} //fim
    
```

## Rotina do evento de chegada do cliente à barbearia

- Enquanto evento de entrada no sistema, este evento é responsável pela geração dos próximos clientes.
- Enquanto evento de início de uma actividade morta ele tem que:

Slide 84

1. Colocar a respectiva entidade no presente estado morto.
2. Caso só exista uma entidade neste estado morto, verificar se a próxima actividade viva pode ter início. Se sim, marcar de imediato o evento de início dessa actividade.

## Rotina do evento de início de corte

```
Inicio_corte()
{//início
  1. passa o cliente de CLIENT_ESPERA para CLIENT_CORTE
  2. passa o barbeiro de BARB_LIVRE para BARB_CORTE
  3. calcula o tempo de corte (tcorte)
  4. Marca(fim_corte, tsim+tcorte) //marca o fim do corte
} //fim
```

Slide 85

## Rotina do evento de início de corte

---

- Enquanto evento de início de uma actividade viva, este evento é responsável por:
  1. Retirar as entidades a envolver nesta actividade dos respectivos estados mortos anteriores e colocá-las no presente estado vivo.
  2. Marcar o evento de fim da presente actividade viva.

Slide 86

## Rotina do evento de fim de corte

---

```
Fim_corte()
{//início
  1. retira cliente de CLIENT_CORTE e elimina-o do sistema
  2. muda o barbeiro de BARB_CORTE para BARB_LIVRE
  //Se existir algum cliente na espera, marca o início do corte
  //para acontecer imediatamente. Caso contrário, não faz nada.
  3. if(#CLIENT_ESPERA > 0)
    {
      Marca(Inicio_corte, tsim) //início do corte imediato
    }
} //fim
```

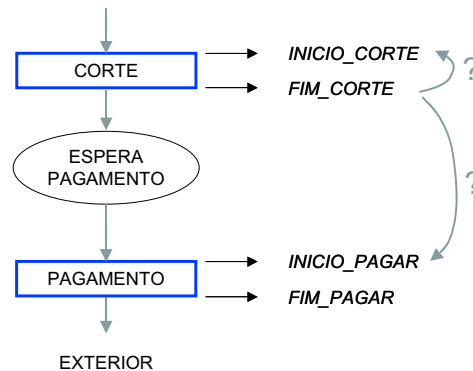
Slide 87

## Rotina do evento de fim de corte

---

### Slide 88

Se existisse outra actividade viva antes da saída para o EXTERIOR (e.g. o pagamento), em vez de se eliminar o cliente dever-se-ia colocá-lo no estado *morto* de acesso à actividade *viva* seguinte e em seguida proceder como na entrada de uma actividade *morta*: não existindo clientes na ESPERA PAGAMENTO, verificar se o PAGAMENTO poderia ser iniciado. Em caso afirmativo marcar-se-ia de imediato tal evento.



## Rotina do evento de fim de corte

---

- Qualquer rotina de um evento que corresponda ao fim de uma actividade viva deverá:
  1. Libertar as entidades envolvidas na actividade viva que finda, colocando-as nos próximos estados mortos.
  2. Verificar se a actividade viva que finda pode ser de novo iniciada no momento. Se sim, marcar de imediato o respectivo evento de início.
  3. Caso só exista uma entidade no estado morto de acesso à próxima actividade viva, e se essa actividade puder ser iniciada no momento, marcar de imediato o respectivo evento de início.

### Slide 89

## Abordagem por eventos – Recolha de dados

---

- Cria-se um novo evento e faz-se executá-lo com a frequência que nos interessar recolher os dados.
- Este evento é também responsável por se marcar a si próprio no futuro, produzindo a frequência de amostragem desejada.

Slide 90

Amostragem()

{//início

1. memoriza número de entidades no EXTERIOR

2. memoriza número de entidades em CLIENT\_ESPERA

3. memoriza número de entidades em CLIENT\_CORTE

4. memoriza número de entidades em BARB\_CORTE

5. calcula o tempo da próxima amostragem (tnext)

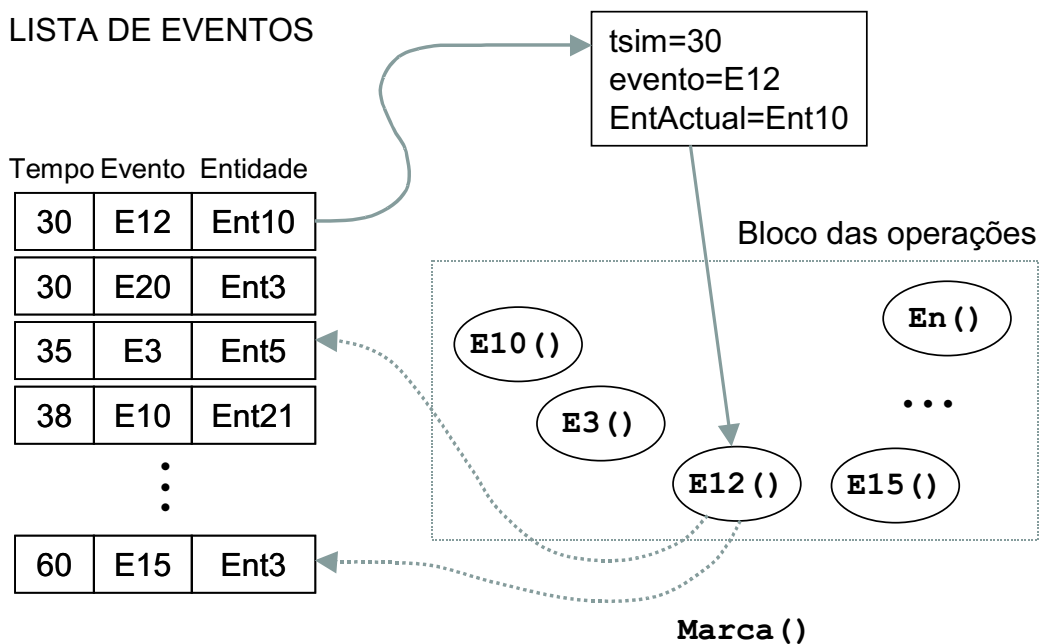
6. Marca(Amostragem, tnext) //marca a próxima amostragem

}//fim

## Abordagem por eventos – o executivo

---

Slide 91



## Abordagem por eventos – o executivo

---

- O executivo começa por retirar da *lista de eventos* o evento que se encontra no topo.
- Há uma entidade associada a cada evento pois uma mesma rotina poderá ser executada sobre entidades temporárias diferentes.
- O executivo coloca o tempo de simulação igual ao tempo desse evento, a entidade actual igual à entidade correspondente e chama a rotina do evento em causa.
- O sistema muda de estado através das acções que compõem a rotina.
- Novos eventos são eventualmente marcados na *lista de eventos* também através destas acções.

Slide 92

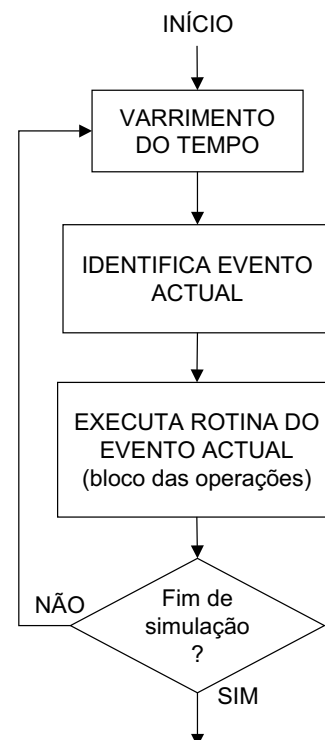
## Esquema geral do executivo na abordagem por eventos

---

```

Executivo()
{//Início
while (fim==FALSE)
  {//VARRIMENTO DO TEMPO:
  {
    1. evento = eventList.GetHead();
    2. tsim = evento.tempo;
    3. EntActual = evento.entidade;
  }
  //IDENTIFICA EVENTO ACTUAL E MANDA EXECUTAR A
  //RESPECTIVA ROTINA
  {
    if (evento==evento1) evento1();
    else if (evento == evento2) evento2();
    else if (evento == evento3) evento3();
    else if (evento == evento4) evento4();
    ...
    else if (evento == eventoN) eventoN();
  }
  }
}
} //fim
    
```

Slide 93



## A função *Schedule()*

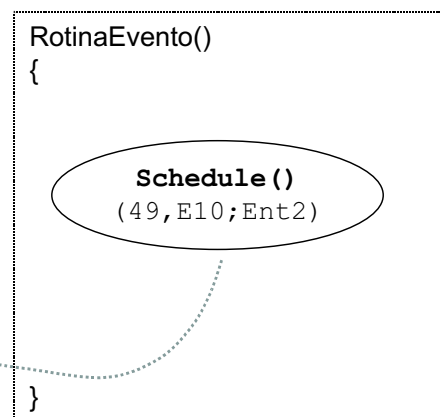
---

- A abordagem por eventos implica a existência de uma função que permita marcar eventos, isto é, inseri-los **por ordem do tempo** na *lista de eventos*.
- A função **Marca()**, ou **Schedule()** como habitualmente surge na literatura, é responsável não só por inserir os eventos, como por manter a lista ordenada.

Slide 94

### LISTA DE EVENTOS

Tempo	Evento	Entidade
30	E20	Ent3
35	E3	Ent5
38	E10	Ent21
⋮		
49	E10	Ent2
60	E15	Ent3



## Algumas notas finais sobre a abordagem por eventos

---

- A abordagem mais popular na última década.
- Eficiente e versátil.
- A execução de cada rotina é absolutamente independente das outras, pelo que a sua posição no bloco das operações em nada influencia o comportamento do modelo.
- É fácil fazer executar os eventos de fim de actividade antes dos de início: basta colocá-los em primeiro lugar na lista de eventos.
- Introduce uma complexidade adicional no simulador: a necessidade da reorganização da lista de eventos, o que é normalmente assegurado pela função *Schedule()*.

Slide 95



## Criação de um modelo de simulação por eventos O problema da travessia de um rio com botes

---

Slide 96

Uma povoação é atravessada por um rio e, para assegurar o transporte diário dos seus habitantes entre as duas margens, há dois botes. A velocidade destes botes é diferente quando carregados ou vazios. Os botes são pequenos e só podem transportar um adulto de cada vez, ou quanto muito um adulto e uma criança. Uma criança sozinha nunca poderá atravessar, devendo aguardar a chegada de um adulto para atravessar com ele. Os botes, ao terminarem o transporte numa margem ficam à espera nessa margem, a menos que haja pedidos pendentes na outra margem. Caso haja pedidos pendentes nas duas margens, o bote dá prioridade aos pedidos da margem onde se encontra.

## O problema da travessia de um rio com botes – continuação

---

Designando as margens como Norte e Sul, a observação do movimento durante 15 minutos resultou nos seguintes dados:

- Tempo de travessia de um bote vazio: 5 minutos
- Tempo de travessia de um bote cheio: 10 minutos
- Chegadas às margens:

Slide 97

Chegadas	Pedido de travessia	Tempo da chegada (min)
Adulto (A1)	N → S	5
Criança (C1)	N → S	6
Adulto (A2)	S → N	10
Adulto (A3)	N → S	12

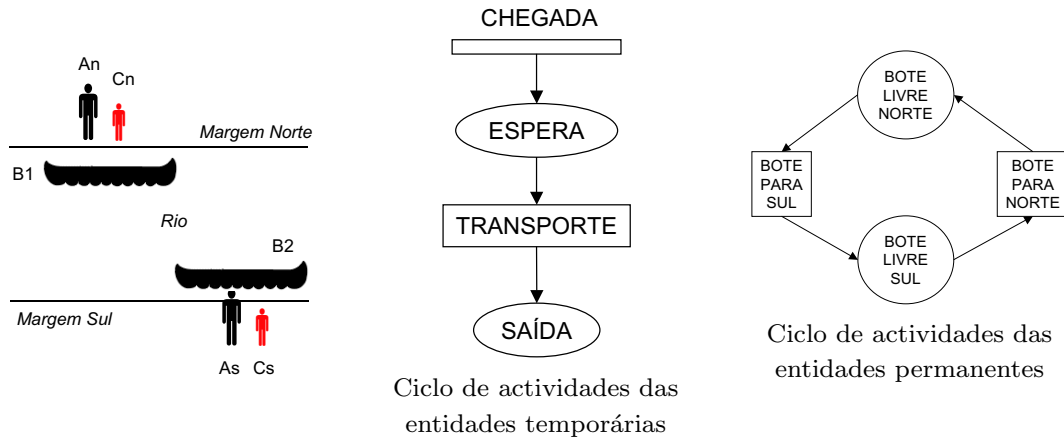
A partir destes dados, e considerando que inicialmente os botes estão um em cada margem, desenvolva um modelo de simulação baseado numa abordagem por eventos e corra a simulação.

## O problema da travessia de um rio com botes

### Identificação das entidades e das actividades

Entidades temporárias – adultos ( $A_n$  e  $A_s$ ) e crianças ( $C_n$  e  $C_s$ ) – e permanentes – botes ( $B_1$  e  $B_2$ )

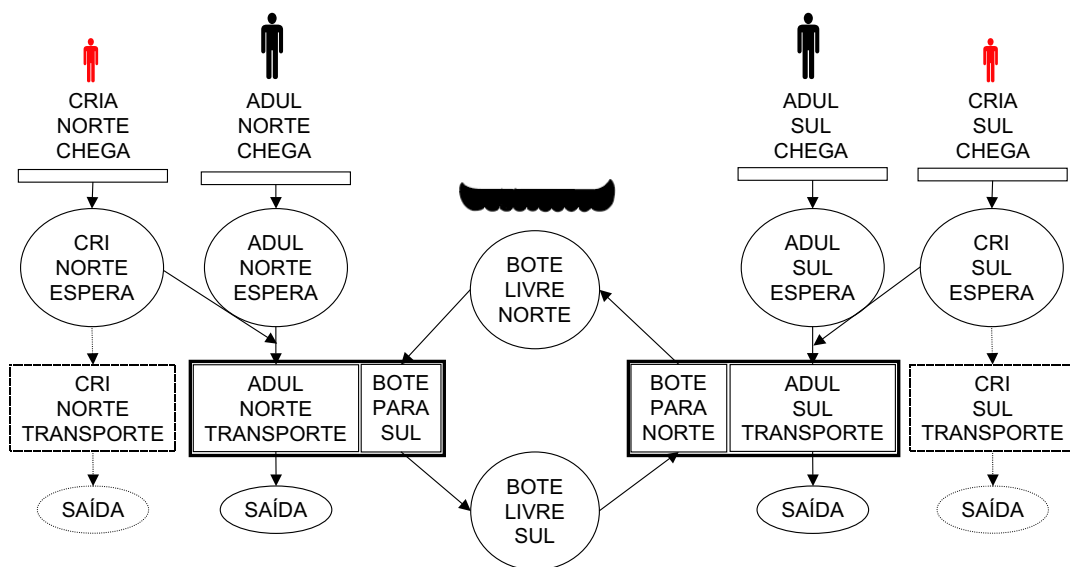
Slide 98



## O problema da travessia de um rio com botes

### Diagrama de estados

Slide 99



## O problema da travessia de um rio com botes Simulação

---

Técnica de avanço de tempo: próximo evento

**tsim = 0**

Slide 100

Lista de eventos		
t	evento	ent.
5	ChAdNS	A1N
6	ChCrNS	C1N
10	ChAdSN	A2S
12	ChAdNS	A3N

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
A1N				C1N						B1	B2
A2S											
A3N											

## O problema da travessia de um rio com botes Simulação – continuação

---

**tsim = 5**

A1N chega e passa para o estado *esp.N*. Porque há um bote no estado *livreN*, de imediato passa para o estado *bote*.

B1 passa para o estado *N→S*.

Marca-se um novo evento na lista de eventos: a chegada à margem sul, 10 minutos depois.

Slide 101

Lista de eventos		
t	evento	ent.
6	ChCrNS	C1N
10	ChAdSN	A2S
12	ChAdNS	A3N
15	fimTpSul	B1

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
A2S			A1N	C1N				B1			B2
A3N											

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 6**

C1N chega, passando para o estado *esp.N*.

Slide 102

Lista de eventos		
t	evento	ent.
10	ChAdSN	A2S
12	ChAdNS	A3N
15	fimTpSul	B1

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
A2S			A1N		C1N			B1			B2
A3N											

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 10**

A2S chega e passa para o estado *esp.S*. Porque há um bote no estado *livreS*, de imediato passa para o estado *bote*.

B2 passa para o estado *S→N*.

Marca-se um novo evento na lista de eventos: a chegada à margem norte, 10 minutos depois.

Slide 103

Lista de eventos		
t	evento	ent.
12	ChAdNS	A3N
15	fimTpSul	B1
20	fimTpNor	B2

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
A3N			A1N		C1N			B1	B2		
			A2S								

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 12**

A3N chega e passa para o estado *esp.N*, porque não há nenhum bote livre para fazer a travessia.

Lista de eventos		
t	evento	ent.
15	fimTpSul	B1
20	fimTpNor	B2

Slide 104

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
	A3N		A1N A2S		C1N			B1	B2		

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 15**

B1 chega à margem sul. Verifica primeiro que não está ninguém à espera na margem sul (estados *esp.S* vazios) e verifica depois que A3N está no estado *esp.N*, pelo que passa para o estado *S→N* e marca o evento de chegada a norte para 5 minutos depois.

A1N sai do sistema.

Lista de eventos		
t	evento	ent.
20	fimTpNor	B2
20	fimTpNor	B1

Slide 105

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
	A3N		A2S		C1N				B2		
									B1		

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 20**

B2 chega à margem norte. Passa ao estado  $N \rightarrow S$  e marca o evento de chegada a sul para 10 minutos depois. A3N e C1N passam ao estado *bote*.

B1 passa ao estado *livreN*.

A2S sai do sistema

Slide 106

Lista de eventos		
t	evento	ent.
30	fmTpSul	B2

Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
			A3N				C1N	B2			
										B1	

## O problema da travessia de um rio com botes

### Simulação – continuação

---

**tsim = 30**

B2 chega à margem sul. Passa ao estado *livreS*.

A3N e C1N saem do sistema.

A lista de eventos fica vazia.

Slide 107

Lista de eventos		
t	evento	ent.

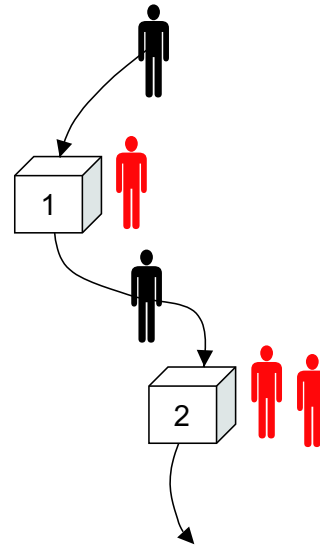
Estados											
Adultos				Crianças				Botes			
ext.	esp.N	esp.S	bote	ext.	esp.N	esp.S	bote	N→S	S→N	livreN	livreS
											B2
										B1	

## Abordagem por processos

---

Criar um modelo:

- Descrever os percursos ou *processos* de cada entidade activa do sistema...
- ... que se obtêm traçando todos os caminhos possíveis da entidade ao longo do seu ciclo de actividades...
- ... que *requerem* ou *libertam* recursos (todos os restantes componentes do sistema são considerados como recursos), que podem estar *ocupados* ou *disponíveis*.
- As entidades activas cujo ciclo de actividades se reduza a dois estados (vivo/ocupado ou morto/disponível) são modeladas como recursos (e.g. o barbeiro no sistema barbearia).

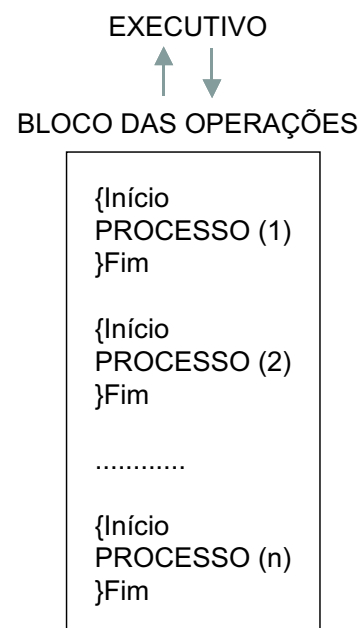


Slide 108

## O bloco das operações – estrutura de um processo

---

- Um processo  $\Leftrightarrow$  bloco de programa/rotina que descreve a vida completa de uma dada **classe de entidades**.
- Cada entidade de uma certa classe que aparece no sistema recebe uma “cópia” do processo associado à sua classe.
- A entidade posiciona-se no início do seu processo e vai tentando evoluir ao longo dele.
- Esta evolução depende da disponibilidade de recursos  $\Rightarrow$  mecanismos para adquirir um recurso (incluindo-se no pedido o tempo de espera por ele) e para libertar recursos (para que fiquem disponíveis para outros processos).



Slide 109

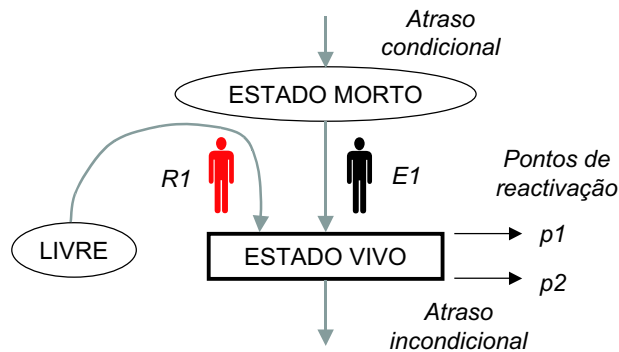


## Atrasos condicionais e atrasos incondicionais

---

Slide 110

- Atrasos condicionais – demoras introduzidas pelas actividades mortas: a entidade deverá esperar até que se encontre no topo do actual estado morto e que se encontrem disponíveis os recursos necessários para a passagem ao próximo estado vivo.



- Atrasos incondicionais - demoras introduzidas pela execução de uma actividade viva: a entidade suspende a sua progressão no processo durante um período de tempo bem determinado.

Pontos de reactivação – pontos onde se deve reavaliar se a entidade pode prosseguir o seu ciclo de actividades.

## Abordagem por processos – o caso da barbearia

---

Slide 111

- Cliente → Entidade  
Barbeiro → Recurso
- Cliente entra no sistema → espera de corte (estado CLIENT\_ESPERA) → atraso condicional: cliente avança novamente no processo quando for o primeiro da fila e o barbeiro estiver livre.
- Estas condições têm que ser continuamente analisadas ao longo da simulação ⇒ cliente é enviado para uma lista de entidades que será inspeccionada em cada ciclo de simulação: **lista de entidades presentes**.
- Cliente a participar na actividade CORTE ⇒ colocação na **lista de eventos futuros**: lista com as entidades que actualmente participam em actividades vivas.



## Rotina do processo cliente

---

Slide 112

```

Processo_cliente() { //início
  1. Calcula chegada do próximo cliente (t=tnext)
  2. Marca(tnext, chegada_cliente, proxCliente)
  -----
  3. Coloca cliente actual em CLIENT_ESPERA
  4. Wait until (barbeiro==LIVRE && cliente no topo de CLIENT_ESPERA)
  -----
  5. Transfere barbeiro para BARB_CORTE
  6. Transfere cliente para CLIENT_CORTE
  7. Calcula tempo de corte (tcorte)
  8. Suspend(tcorte) //Espera durante o tempo de corte
  -----
  9. Coloca barbeiro em BARB_LIVRE //Barbeiro fica livre
  10. Retira cliente de CLIENT_CORTE e elimina-o do sistema
} //fim

```

## Rotina do processo cliente

---

Slide 113

- Sempre que um cliente chega ao sistema é-lhe atribuída uma réplica deste processo que ele começa a percorrer.
- Quando o cliente é colocado em CLIENT\_ESPERA (atraso condicional) é transferido para a *lista de eventopresentes*.
 

}	procedimento Request(barbeiro) usado nalgumas linguagens de simulação.
---	---
- Este é também um *ponto de reactivação* com o início do corte.
- Outro ponto de reactivação do processo é o fim do corte – procedimento Release(barbeiro)

Nalgumas linguagens o procedimento Release() cria as entidades caso elas não existam ainda.

## Separação dos processos em dois...

---

- ... processo de entrada da entidade no sistema (mais simples)
- e o processo de evolução da entidade no sistema (o verdadeiro “processo”).

Slide 114

```

Processo_chegaCliente() { //início
    While (SIMULAR==TRUE)
        {
            1. cria novo cliente no sistema. //Release(cliente)
            2. Calcula chegada do próximo cliente (t=tnext)
            3. Suspend(tnext-tsim) //note-se que tsim=tempoActual
        }
} //fim

```

## Separação dos processos em dois...

---

Slide 115

```

Processo_cliente() { //início
    1. Coloca cliente actual em CLIENT_ESPERA
    2. Wait until (barbeiro==LIVRE && cliente no topo de CLIENT_ESPERA)
        //Entra para o corte:-----
    3. Transfere barbeiro para BARB_CORTE
    4. Transfere cliente para CLIENT_CORTE
    5. Calcula tempo de corte (tcorte)
    6. Suspend(tcorte) //Espera que acabe o corte
        //Fim do corte:-----
    7. Coloca barbeiro em BARB_LIVRE //Barbeiro fica livre
    8. Retira cliente de CLIENT_CORTE e elimina-o do sistema
} //fim

```

## Abordagem por processos – Recolha de dados

---

- Cria-se um novo processo que corre em paralelo com os outros processos.

```
Processo_Amostragem() { //início
    1. memoriza número de entidades no EXTERIOR
    2. memoriza número de entidades em CLIENT_ESPERA
    3. memoriza número de entidades em CLIENT_CORTE
    4. memoriza número de entidades em BARB_CORTE
    5. calcula o tempo da próxima amostragem (tnext)
    6. Suspend(tnext-tsim) //espera a próxima amostragem
} //fim
```

Slide 116

## Abordagem por processos – o executivo

---

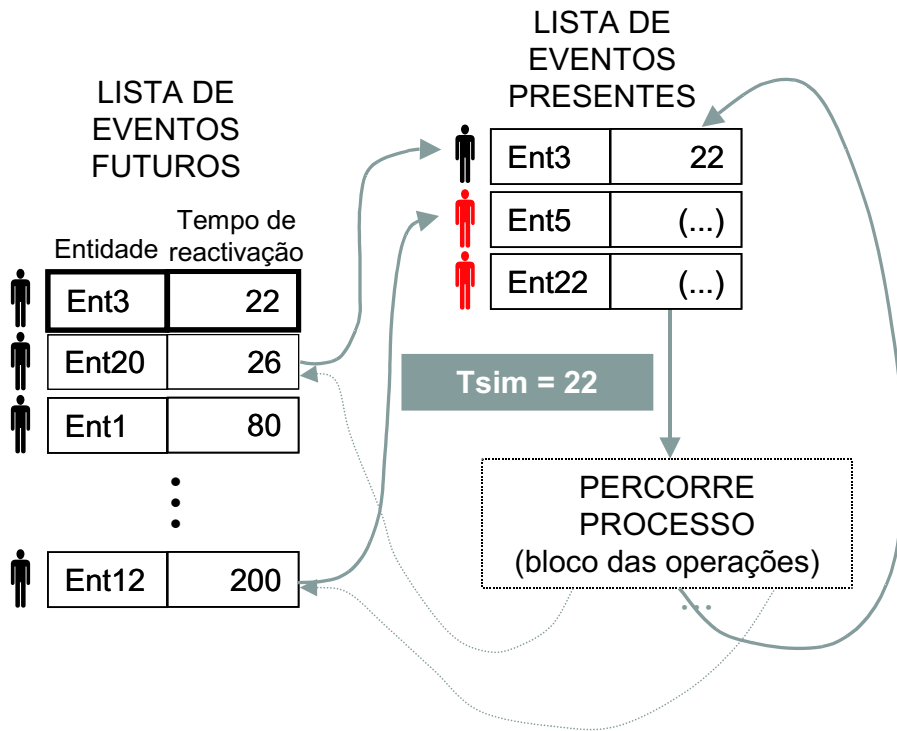
- Os pontos de reactivação podem corresponder a *atrasos condicionais* ou a *atrasos incondicionais*  $\Rightarrow$  o executivo precisa de distinguir as acções a realizar em cada um dos casos...
- ... considerar em cada avanço do tempo da simulação dois tipos de acontecimentos: os que ocorrerão no presente e os que, de certeza, só ocorrerão no futuro.
- – *Lista de eventos presentes*: contém entidades que no presente momento da simulação estão envolvidas em actividades mortas ou em actividades vivas que terminem nesse momento.
- – *Lista de eventos futuros*: contém entidades implicadas nas actividades no momento vivas, organizadas por ordem cronológica de finalização.

Durante o tempo de permanência no sistema as entidades activas dividem o seu tempo por estas duas filas.

Slide 117

### Abordagem por processos – o executivo

Slide 118



### Abordagem por processos – o executivo

Slide 119

- Para determinar o tempo actual da simulação, o executivo começa por retirar da *lista de eventos futuros* a entidade ou entidades com menor *tempo de reactivação*: (Ent3, tempo = 22).
- O tempo de simulação é colocado igual a 22 e a entidade Ent3 colocada na *lista de eventos presentes*. Nessa lista estão também representadas (a vermelho) as entidades sujeitas a atrasos condicionais, para as quais não tem sentido falar em *tempo de reactivação*.
- O *executivo* manda continuar o processo de cada entidade da *lista de eventos presentes* até que nele seja encontrada uma nova condição de paragem: um novo atraso *condicional* ou *incondicional*.
- Se o atraso for *condicional* é mantido na *lista de eventos presentes*. Se for *incondicional* a entidade é transferida para a *lista de eventos futuros*, já com o *tempo de reactivação* actualizado.

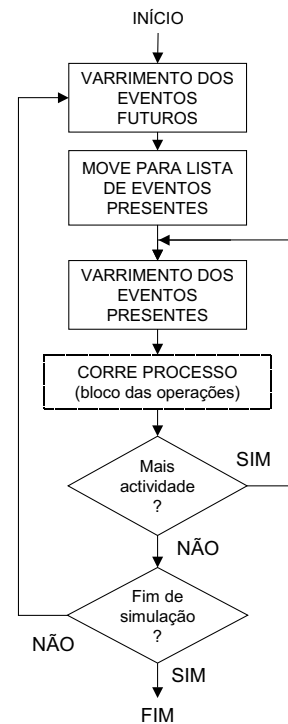
## Esquema geral do executivo na abordagem por processos

Slide 120

```

Executivo()
{ //Início
while (simular==TRUE) // Enquanto for para simular
  { //VARRIMENTO DOS EVENTOS FUTUROS
  1. Percorre listaEventosFuturos e detecta
    as entidades com menor tempo de reactivação
    (tmínimo)
  2. tsim = tmínimo;
  3. Transfere essas entidades para a
    listaEventosPresentes

  for (i=1 to #listaEventosPresentes) // percorre entidades
  {
  //VARRIMENTO DOS EVENTOS PRESENTES
  4. EntActual = próxima da listaEventosPresentes
  //CORRE PROCESSO
  5. EntActual.Processo(tsim);
  }
  }
} //fim
    
```



## Último slide

Simulação não é uma técnica de optimização: é uma técnica para estimar medidas de desempenho dos sistemas modelados.

Na medida em que permite avaliar e comparar diferentes configurações ou parametrizações para os sistemas, ou mesmo comparar sistemas alternativos, pode ser utilizada para apoiar processos de tomada de decisões.

Slide 121

## Bibliografia

---

- António E.S. Carvalho Brito e J. Manuel Feliz Teixeira (2001). *Simulação por computador. Fundamentos e implementação de código em C e C++*. Publindústria.
- Handy A. Taha (1987). *Operations Research: an introduction*. Prentice Hall.
- Ravindram, Philips and Solberg (1987). *Operations Research: principles and practice*. Wiley.
- Hillier and Lierberman (2001). *Introduction to Operations Research*. Mc Graw-Hill.
- Joseph G. Ecker and Michael Kupferschmid (1988). *Introduction to Operations Research*. Wiley.

Slide 122