

## 5. Simulação de Processos (de Eventos em Tempo Discreto)

Ref.: Winston, Operations Research, 4.ed., cap.21

Seções: 21.1, 21.2, 21.6 e 21.7

3

### Introdução

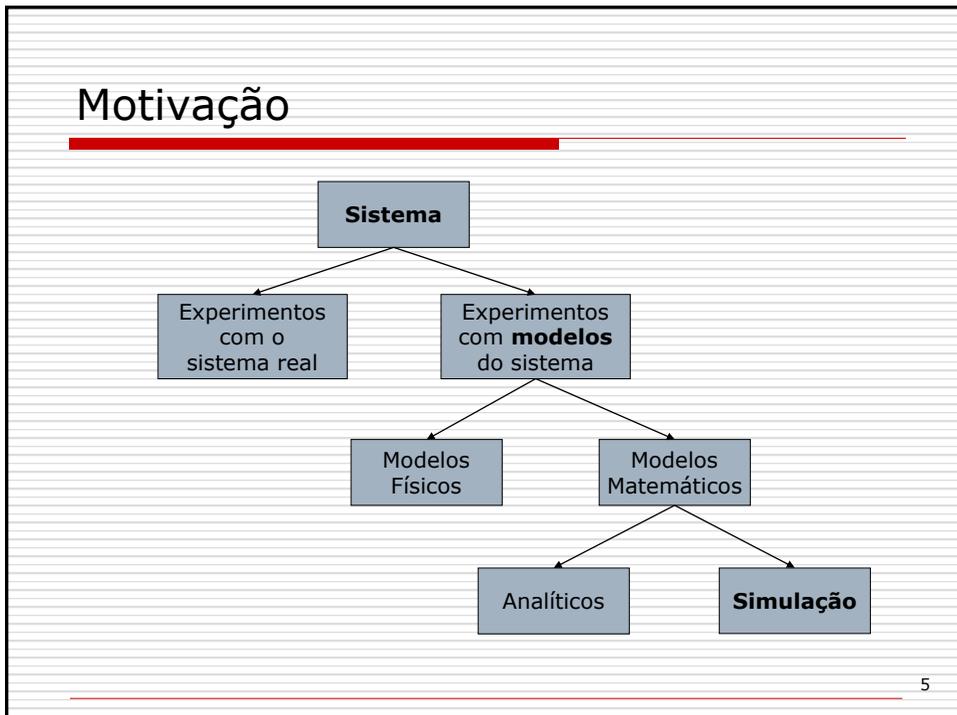
■ **Simulação** é uma técnica da P.O. muito útil para análise de sistemas complexos. Pode ser definida como estudo da dinâmica de sistemas reais, a partir da construção de modelos (computacionais) de simulação.

■ **Método:**

1. Definição do Problema e Objetivos
2. Coleta de Dados e Modelagem
3. Experimentação
4. Análise dos Resultados
5. Decisões

4

4



5

- ## Modelos de Simulação
- Descritivos
  - Estáticos / Dinâmicos
  - (Tempo) Discretos / Contínuos
  - Determinísticos / Estocásticos
  
  - Simulação de Monte Carlo (MC)
  - Simulação de Eventos Discretos (DES)
  - Simulação Dinâmica de Sistemas (DS)
  - Simulação baseada em Agentes (ABS)
- 6

6

## Por que simulação?

- Necessidade de estudar um sistema existente ou em projeto, para:
  - Melhor compreender sua dinâmica e
  - Dimensionar e avaliar novas configurações (melhoria).
  
- Utiliza-se a simulação quando:
  - o sistema real não pode ser estudado diretamente (custo, tempo, riscos etc) e
  - outras técnicas (analíticas) não são adequadas (muitas simplificações)

7

7

## Vantagens e Desvantagens

- Vantagens:
  - **Flexibilidade** – permite a solução de problemas complexos para os quais não há solução analítica simples
  - Menor grau de abstração e, por isso, maior **facilidade de compreensão** e aceitação que os demais modelos de P.O.
  
- Desvantagens:
  - Necessidade de **software**
  - Análise **estatística** dos resultados
  - Modelo Descritivo (não Normativo)

8

8

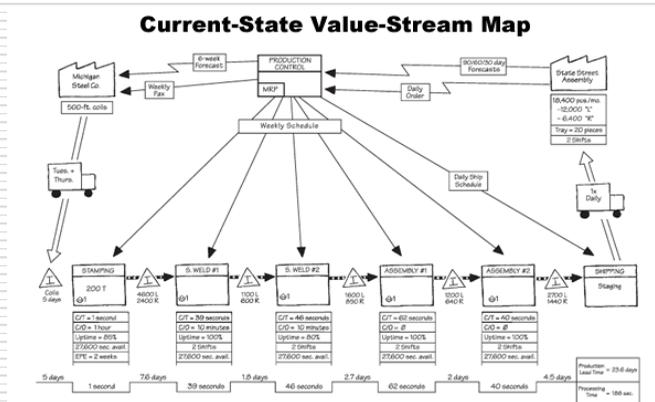
## Áreas de Aplicação

- Produção
  - Logística
  - Serviços
  - Pesquisa Científica
  - Ciências da Computação
  - Aplicações Militares...
- <http://www.wintersim.org/index.htm>

9

9

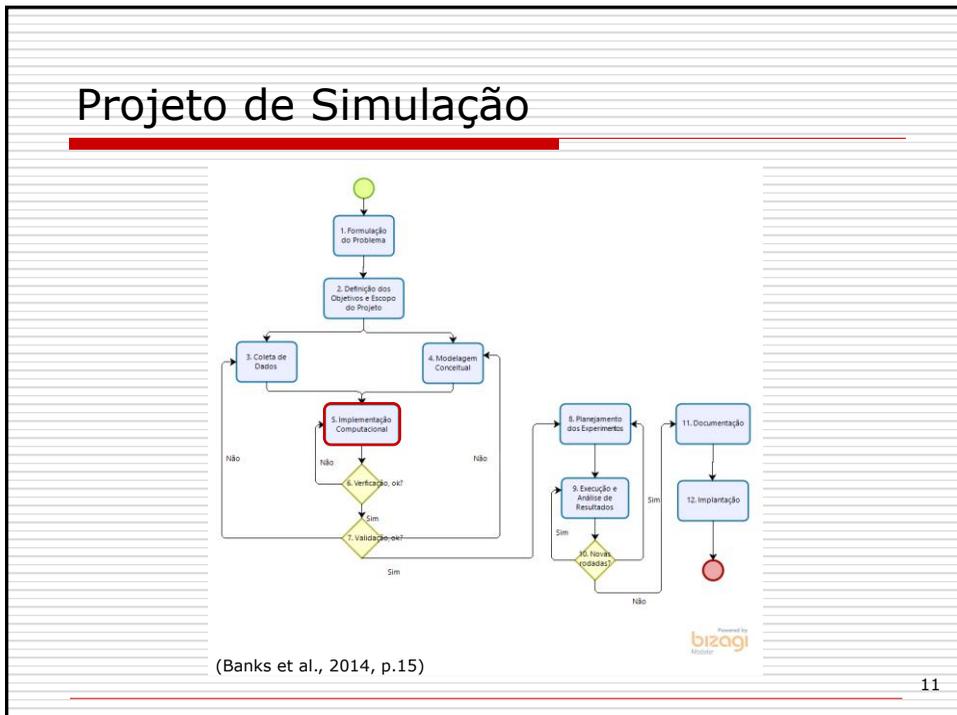
## Simulação de Processos: Mapeamento de Processos



<https://www.lean.org/lexicon/value-stream-mapping>

10

10



11

## 21.1 Terminologia

- Um conceito fundamental na Teoria da Simulação é o de **sistema**, denominação usual do objeto de análise dos projetos de simulação.
- Um sistema consiste de um **conjunto de partes** que atuam e interagem, provocando resultados lógicos decorrentes (agentes, recursos e processos)
- O estado do sistema é determinado por um conjunto de variáveis, denominadas **variáveis de estado**, que descrevem a situação do sistema em cada instante  $t$ .
- Por **exemplo**, o número de jobs na fila, em um dado instante, e o status da máquina (ocupada ou livre) são variáveis de estado do sistema.

12

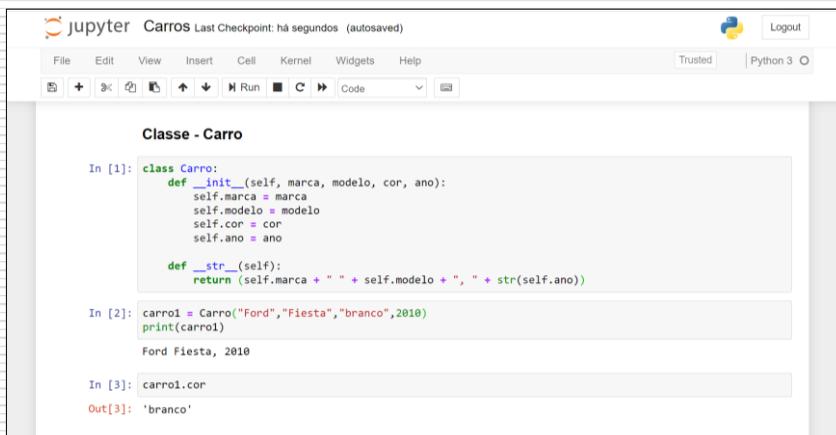
12

- Define-se **evento** qualquer acontecimento que provoca uma mudança de estado do sistema.
- Por exemplo, em um sistema de filas, a **chegada** e **saída** de um job são eventos que alteram o estado do mesmo.
- Em simulação, cada **entidade** pode apresentar propriedades específicas, denominadas **atributos**.
- Por exemplo, cada entidade **job** possui atributos próprios, como **prioridade** e **prazo** de entrega.
- Os modelos de simulação aplicam a lógica de **programação orientada a objetos (OOP)**.

13

13

## OOP: Object Oriented Programming



```

jupyter Carros Last Checkpoint: há segundos (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
+ - < > Run C Code

Classe - Carro

In [1]: class Carro:
        def __init__(self, marca, modelo, cor, ano):
            self.marca = marca
            self.modelo = modelo
            self.cor = cor
            self.ano = ano

        def __str__(self):
            return (self.marca + " " + self.modelo + " " + str(self.ano))

In [2]: carro1 = Carro("Ford","Fiesta","branco",2010)
        print(carro1)
        Ford Fiesta, 2010

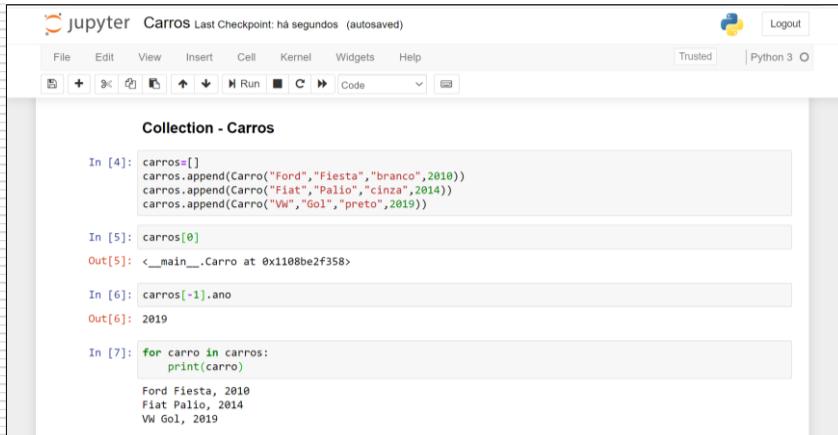
In [3]: carro1.cor
Out[3]: 'branco'

```

14

14

## Classes, Objetos e Coleções



```
Collection - Carros

In [4]: carros=[]
carros.append(Carro("Ford","Fiesta","branco",2010))
carros.append(Carro("Fiat","Palio","cinza",2014))
carros.append(Carro("VW","Gol","preto",2019))

In [5]: carros[0]
Out[5]: <__main__.Carro at 0x1108be2f358>

In [6]: carros[-1].ano
Out[6]: 2019

In [7]: for carro in carros:
print(carro)

Ford Fiesta, 2010
Fiat Palio, 2014
VW Gol, 2019
```

15

15

## Simulação por Eventos Discretos

- **Simulação por eventos discretos:** a dinâmica da simulação é caracterizada por uma sucessão de eventos.
- Todas as informações a respeito dos eventos são preservadas em uma **tabela de eventos**.
- O tempo na simulação é controlado, utilizando uma variável denominada **relógio** da simulação
- Processado um evento, o relógio da simulação avança até o instante de ocorrência do próximo evento na **tabela de eventos futuros**.
- Esta abordagem é denominada **mecanismo de avanço do tempo pelo próximo evento**.

16

16

- Os avanços no relógio, correspondentes aos **saltos** de um evento para outro, tendem a ser bastante irregulares.
- O processamento de um evento implica na atualização das variáveis de estado e na programação de **novos eventos futuros** decorrentes.
- **Exemplo**, o fim de um atendimento pode dar origem, se houver algum job na fila, a um novo atendimento.
- Os estudos de simulação podem concentrar-se no regime **transitório** (condições iniciais) ou **permanente** (longo prazo).
- Exemplos de **aplicação**: simulação de uma fábrica, um terminal de carga, um pronto socorro etc.

17

17

## Tabela de Eventos Futuros

*sort*
*pop*
*append*
*sort*
*pop*

Tempo	Evento	Tipo																														
0,279	E1	T3	0,132	E5	T3	0,203	E4	T2	0,203	E4	T2	0,203	E4	T2	0,279	E1	T3	0,279	E1	T3	0,279	E1	T3									
0,470	E2	T2	0,203	E4	T2	0,279	E1	T3	0,285	E6	T1	0,285	E6	T1	0,470	E2	T2															
0,593	E3	T3	0,279	E1	T3	0,470	E2	T2	0,470	E2	T2	0,470	E2	T2	0,593	E3	T3															
0,203	E4	T2	0,470	E2	T2	0,593	E3	T3	0,593	E3	T3	0,593	E3	T3	0,593	E3	T3	0,593	E3	T3												
0,132	E5	T3	0,593	E3	T3				0,285	E6	T1	0,285	E6	T1	0,593	E3	T3	0,593	E3	T3												

*insert*

Tempo	Evento	Tipo	Tempo	Evento	Tipo
0,203	E4	T2	0,203	E4	T2
0,279	E1	T3	0,279	E1	T3
0,470	E2	T2	0,285	E6	T1
0,593	E3	T3	0,470	E2	T2
			0,593	E3	T3

18

18

## 21.2 Exemplo: Simulação de uma Fila G/G/1 (Winston, 2004)

---

- Este exemplo trata de uma fila G/G/1 com os seguintes pressupostos:
  - a **população** de clientes que gera a demanda (chegadas) é **infinita**
  - a **capacidade** da fila é **ilimitada** e os clientes são atendidos por um **único servidor**, pela ordem de chegada (FCFS)
  - as **chegadas** são individuais (um job por vez), em intervalos de tempo aleatórios, independentes e estacionários
  - todo cliente é atendido (não há desistências), com tempos de **atendimento** aleatórios, independentes e estacionários
  - após o atendimento, os jobs deixam o sistema (atendimento em estágio único)

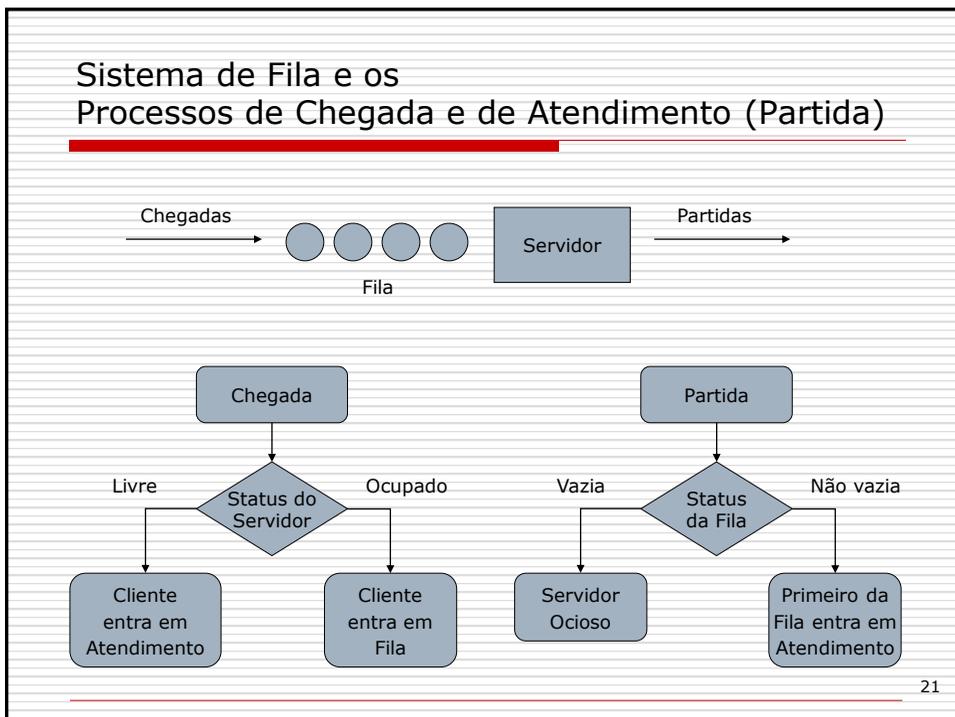
19

19

- 
- Neste exemplo, o estado do sistema é dado por:
    - **status do servidor** - livre ou ocupado;
    - **número de clientes** na fila;
    - **instante da próxima chegada**; e
    - **instante da próxima partida**, se houver cliente em atendimento.
  - No exemplo, há dois eventos básicos: **chegada** e **partida** (término de atendimento) dos clientes.

20

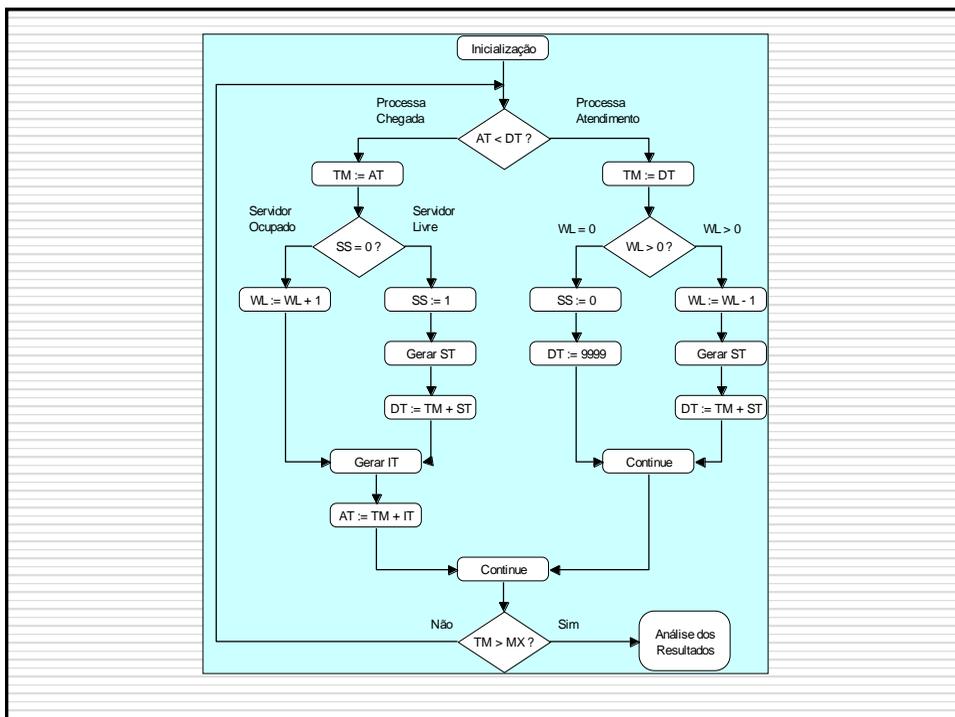
20



21

- Para apresentar o modelo, definem-se as seguintes variáveis:
    - $TM$  = clock time of the simulation
    - $AT$  = scheduled time of the next arrival
    - $DT$  = scheduled time of the next departure
    - $IT$  = interarrival time (  $AT = TM + IT$  )
    - $ST$  = service time (  $DT = TM + ST$  )
    - $SS$  = status of the server (0=idle, 1=busy)
    - $WL$  = length of the waiting line
    - $MX$  = length (in time units) of a simulation run
- 22

22



23

Mostrar um código em ipynb

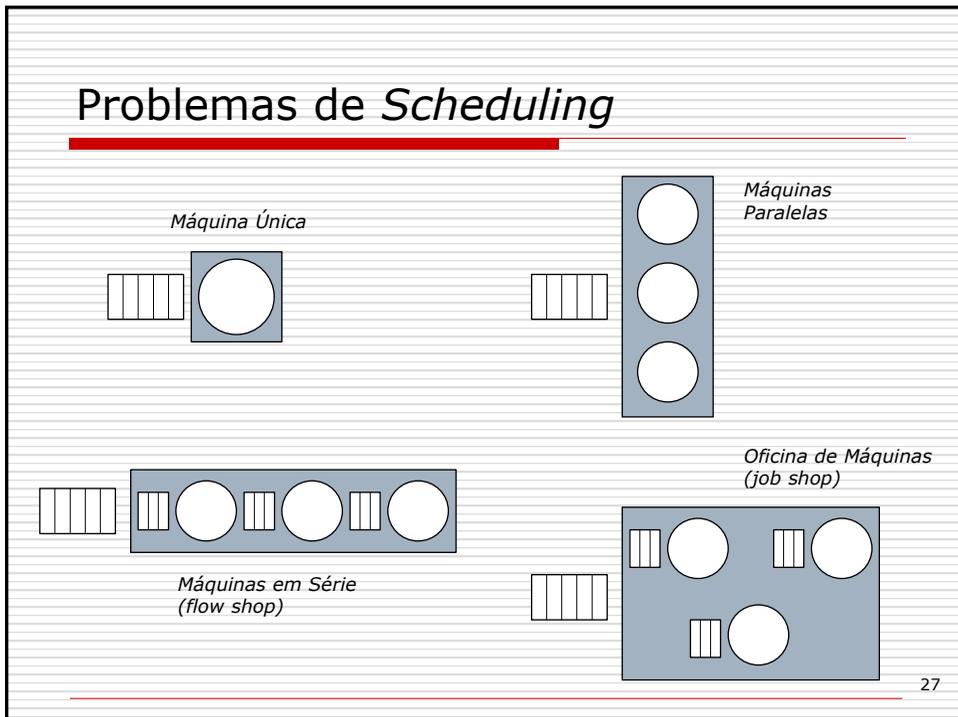
## G/G/3 Python

---

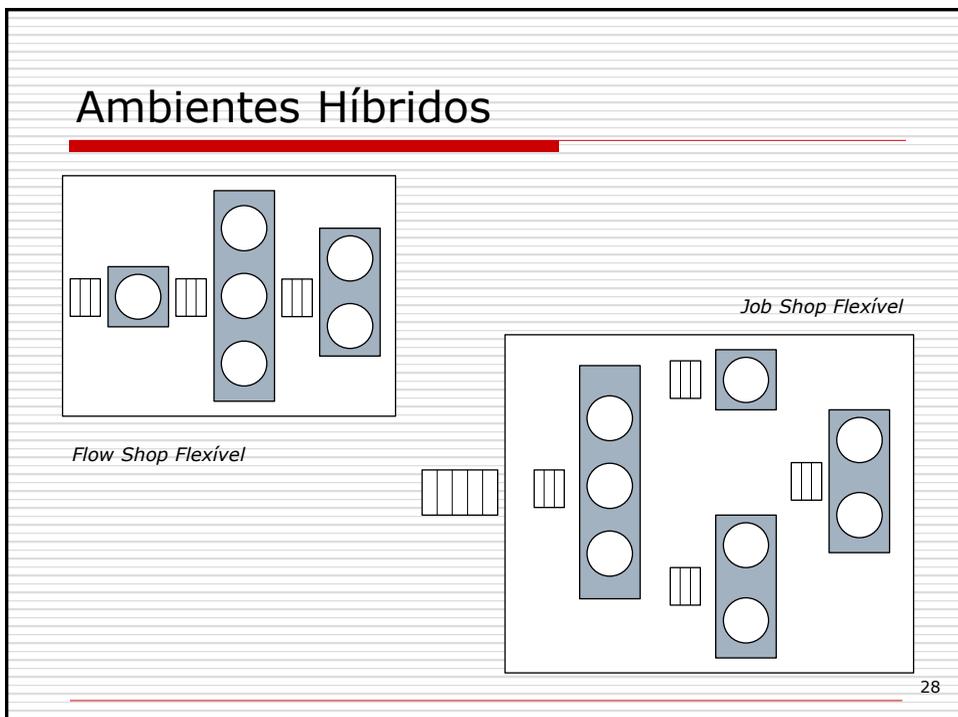
<pre> class Job:     pass  class Evento:     pass  def procChegada(evento):     return tnow  def procPartida(evento):     return tnow  """ Bloco principal """  tnow=0 # relógio da simulação  evento=Evento('primeira chegada') eventos=[evento]  while len(eventos)&gt;0:     evento=eventos.pop(0)     if evento.tipo==0:         tnow = procChegada(evento)     else:         tnow = procPartida(evento)         </pre>	<pre> 0.807 chegada do job 0 0.807 início de serviço do job 0 no máquina 0 2.039 chegada do job 1 2.039 início de serviço do job 1 no máquina 1 5.027 chegada do job 2 5.027 início de serviço do job 2 no máquina 2 7.763 saída do job 0 do máquina 0 12.355 saída do job 2 do máquina 2 13.729 chegada do job 3 13.729 início de serviço do job 3 no máquina 0 18.189 saída do job 1 do máquina 1 19.347 chegada do job 4 19.347 início de serviço do job 4 no máquina 1 22.784 saída do job 3 do máquina 0 25.756 chegada do job 5 25.756 início de serviço do job 5 no máquina 0 29.283 chegada do job 6 29.283 início de serviço do job 6 no máquina 2 30.303 chegada do job 7 33.839 chegada do job 8 34.288 saída do job 4 do máquina 1 34.288 início de serviço do job 7 no máquina 1 40.852 chegada do job 9 43.693 saída do job 6 do máquina 2 43.693 início de serviço do job 8 no máquina 2 44.467 saída do job 7 do máquina 1 44.467 início de serviço do job 9 no máquina 1 51.842 saída do job 9 do máquina 1 58.572 saída do job 8 do máquina 2 87.543 saída do job 5 do máquina 0         </pre>
---	---

24





27



28



## Softwares de Simulação

---

- Arena
- Simul8
- **AnyLogic**
- **Python SimPy**
- ProModel
- FlexSim
- Simio
- ...

<http://www.orms-today.org/surveys/Simulation/Simulation.html>

31

31

## Requisitos do Software de Simulação

---

- Possibilidade de uso interativo
- Boa Interface (user-friendly) e de fácil compreensão
- Estrutura de programação modular, com módulos padrões pré-programados.
- Permitir incorporar nos modelos rotinas e regras de decisão específicas elaborados pelo próprio usuário.
- Flexibilidade para representação de processos produtivos específicos (exemplo, célula de manufatura).

32

32

## Requisitos do Software (cont.)

- Capacidade de rastreamento de fluxos de materiais.
- Índices estatísticos padrões como tempos de fluxo, nível de utilização e tempos de fila.
- Dispor de recursos para análise tanto de dados de entrada como dos resultados da simulação.
- Animação gráfica para ilustrar os fluxos de materiais e operações dentro sistema em simulação.
- Facilidade para depuração do modelo (erros de programação)

33

33

## 21.7 Análise Estatística da Simulação

- Na simulação probabilística, os resultados sempre terão **variações aleatórias**, portanto, demandam uma análise estatística.
- As principais **medidas de desempenho** devem ser expressas na forma de **intervalos de confiança**.
- A determinação de intervalos de confiança, no entanto, exige certos cuidados pois, em simulação, raramente os resultados são independentes (amostras aleatórias).
- Além do problema da **autocorrelação**, deve-se tomar cuidado com o efeito das **condições iniciais** e do **tempo de simulação** nos resultados finais.

34

34

## Período de Aquecimento (transitório)

- O período inicial da simulação é denominado **regime transitório** ou **período de aquecimento**.
- Há duas medidas para conter a influência do transitório no resultado da simulação.
- A primeira é utilizar uma condição inicial próxima de uma condição representativa de regime estacionário.
- A segunda é rodar a simulação e descartar um período inicial no cálculo das estatísticas (“*warm up*” ou “volta de aquecimento” da simulação).
- Como cada modelo é diferente, não há uma regra geral para determinar qual a duração razoável da “volta de aquecimento”. (Ref.: Law and Kelton, 1991.)

35

35

## Critério de Parada

- Há basicamente dois critérios de parada: por evento determinado (“**terminating simulations**”) ou até entrar em regime permanente (“**steady-state simulations**”).
- No primeiro caso, encerra-se o experimento no instante de ocorrência de um determinado evento. Por exemplo, quando não houver mais jobs para atender.
- No segundo, roda-se o modelo por um tempo suficientemente longo de tal forma que as condições iniciais tenham pouco ou nenhum impacto nos resultados (em regime).
- A análise estatística pode ser feita a partir de múltiplas replicações independentes ou uma única corrida por um tempo longo em regime. A escolha depende da natureza do problema.

36

36

## Exemplo: caso *Cabot* (Winston, p.1173)

- Considere-se o caso com 11 terminais e apenas 10 replicações da simulação com parada por tempo determinado (iguais nas 10 replicações).
- Para cada replicação, tem-se uma média de terminais em operação (indicador de disponibilidade) ao longo da simulação.
- A partir da amostra deste indicador (n=10 replicações), tem-se uma média global de 9,331 e variância 0,018.
- Determina-se, então, o intervalo com 95% de confiança do valor médio do indicador por:

$$9,331 \pm 2,26 \sqrt{\frac{0,0180}{10}} = 9,331 \pm 0,096$$

- Neste exemplo, verifica-se que a análise dos resultados é relativamente simples quando se adotam múltiplas replicações com critério de parada predefinido.

37

37

## Próxima aula – Python / SimPy

- Lab de Python / SimPy
- IDE: Jupyter Notebook
- Instalar: SimPy



38

38