

TP 547- Princípios de Simulação de Sistemas de Comunicação

Prof. Samuel Baraldi Mafra



Bibliografia

- Chwif, L.; Medina, A. C. Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações. 4^o edição. São Paulo: Elsevier Brasil, 2014. 320 p. (Capítulo 3)
- Fishman, George S. Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media, 2013. (Capítulo 2)
- Banks, Jerry et al. Discrete-event system simulation. Pearson, 2005. (Capítulo 3)

Componentes do modelo

Modelo conceitual: Componentes

- **Sistema** Uma coleção de entidades (por exemplo, pessoas e máquinas) que interagem juntas ao longo do tempo para realizar um ou mais objetivos.
- **Modelo** Uma representação abstrata de um sistema, geralmente contendo relações estruturais, lógicas ou matemáticas que descrevem uma sistema em termos de estado, entidades e seus atributos, conjuntos, processos, eventos, atividades e atrasos.
- **Estado do sistema** Uma coleção de variáveis que contém todos os informações necessárias para descrever o sistema a qualquer momento.

Modelo conceitual: Componentes

- **Entidade** Qualquer objeto ou componente no sistema que requer representação explícita no modelo (por exemplo, um servidor, um cliente, um máquina).
- **Atributos** As propriedades de uma determinada entidade (por exemplo, a prioridade de um cliente em espera, o encaminhamento de um trabalho através de uma loja de trabalho).
- **Lista** Uma coleção de (permanentemente ou temporariamente associada) entidades, ordenadas de alguma forma lógica (como todos os clientes atualmente em uma linha de espera, ordenada por primeiro a chegar, primeiro a ser servido ou prioridade).

Modelo conceitual: Componentes

- **Evento** Uma ocorrência instantânea que altera o estado de um sistema (como a chegada de um novo cliente).
- **Aviso de evento** Um registro de um evento para ocorrer no momento atual ou algum tempo futuro, juntamente com quaisquer dados associados necessários para executar o evento; no mínimo, o registro inclui o tipo de evento e o hora do evento.
- **Lista de eventos** Uma lista de avisos de eventos para eventos futuros, ordenada por hora de ocorrência; também conhecida como a lista de eventos futuros (FEL).

Tempo

- **Atividade** Um tempo de duração especificado (por exemplo, tempo médio de serviço ou tempo entre chegadas), que é conhecido quando começa a simulação.
- **Atraso** Um tempo de duração indeterminado, não especificado a priori (por exemplo, o atraso de um cliente em um fila de espera).
- **Clock** Uma variável que representa o tempo simulado.

Atividade

- A duração de uma atividade é computável a partir de sua especificação no instante em que começa. Sua duração não é afetados pela ocorrência de outros eventos (a menos que, como é permitido por alguns pacotes de simulação, o modelo contém lógica para cancelar um evento).
- Para acompanhar as atividades e seu tempo de conclusão esperado, quando uma duração de atividade começa, um aviso de evento é criado tendo um tempo de evento igual ao tempo de conclusão da atividade.
 - Se o tempo simulado atual for $CLOCK = 100$ minutos e um tempo de execução da tarefa é de exatamente 5 minutos, então um um aviso de evento é criado especificando o tipo de evento e o tempo de termino do evento ($100 + 5 = 105$ minutos).

Atraso

- Em contraste com uma atividade, a duração de um atraso não é especificada antes da simulação, mas é determinado pelas condições do sistema.
 - Por exemplo, o atraso de um cliente em uma fila de espera pode depender do número e da duração de serviço de outros clientes no futuro em linha, bem como a disponibilidade de servidores e equipamentos.
- Um atraso costuma ser chamado de **espera condicional**, enquanto uma atividade é chamada de **espera incondicional**.

Atraso

- A conclusão de uma atividade é um evento, geralmente chamado de **evento primário**, que é gerenciado colocando um aviso de evento no FEL.
- Em contraste, os atrasos são gerenciados colocando a entidade associada em outra lista, por exemplo uma lista de processos na fila de espera.
- A conclusão de um atraso é chamada de evento condicional ou **evento secundário**, mas esses eventos não são representados por evento avisos, nem aparecem no FEL.
- Os sistemas considerados aqui são dinâmicos.

- A definição dos componentes do modelo fornece uma descrição estática do modelo.
- Uma simulação de evento discreto produz uma sequência de instantâneos do sistema (ou imagens do sistema) que representam a evolução do sistema ao longo do tempo.

Modelo Conceitual

Uma imagem de um determinado momento ($\text{CLOCK} = t$) não inclui apenas o estado do sistema no tempo t , mas também uma lista (FEL) de todas as atividades atualmente em andamento e quando cada uma dessas atividades terminará, o status de todas as entidades e membros atuais de todos os conjuntos, além da valores de estatísticas e contadores cumulativos que serão usados para calcular estatísticas de resumo no final da simulação.

CLOCK	System state	Entities and attributes	Set 1	Set 2	...	Future event list (FEL)	Cumulative statistics and counters
t	(x, y, z, \dots)					$(3, t_1)$ – Type 3 event to occur at time t_1 $(1, t_2)$ – Type 1 event to occur at time t_2 . . .	

Modelo Conceitual: World views

O modelador pode adotar diferentes visões de mundo ou orientações para desenvolvimento de um modelo, sendo as três mais conhecidas:

- **Agendamento de eventos** (*event-scheduling*)
 - A simulação se baseia em **eventos** e seu efeito no estado do sistema. Usa a tabela de eventos futuros para o encadeamento das fases da simulação.
- **Interação de processos** (*process-interaction*)
 - O modelo é definido em termos de **entidades** ou objetos e seu **ciclo de vida** de uma entidade. Simula os diferentes processos em paralelo como em um sistema operacional de computador.
 - Tem apelo intuitivo e permite descrever o processo em termos de construções de blocos ou redes de alto nível.
 - O agendamento de eventos está oculto.
- Essas duas abordagens usam um avanço de tempo variável; isto é, quando todos eventos e mudanças de estado do sistema ocorreram em um instante de simulado, o relógio de simulação é adiantado para a hora do próximo evento iminente no FEL.

- **Varredura de atividade** (Activity-scanning) usa um incremento de tempo fixo e é baseado em regras para decidir qual atividade pode começar em cada ponto tempo simulado.
Lento!

- A cada ciclo de clock, as condições de cada **atividade** são verificadas e se eles são verdadeiros, a atividade correspondente começa.
- É adequado para sistemas pequenos.
- Melhoria: abordagem de **três fases**.

Agendamento de eventos

- A sequência de ações que um simulador (ou linguagem de simulação) deve executar para avançar o relógio e criar um novo instantâneo do sistema é chamado de **algoritmo de agendamento de eventos / avanço de tempo**.
- O mecanismo para avançar o tempo de simulação e garantir que todos os eventos ocorram em ordem cronológica correta tem como base a lista de eventos futuros (FEL).

- A qualquer momento t , a FEL contém todos os eventos futuros previamente estabelecidos e seus tempos de evento associados (chamados t_1, t_2, \dots). A FEL é ordenada por hora do evento, o que significa que os eventos estão organizados cronologicamente:

$$t < t_1 < t_2 < t_3 < \dots < t_n$$

- O tempo t é o valor atual do tempo simulado.
- O evento associado ao tempo t_1 é chamado de **evento iminente**.

- Após a realização da imagem instantânea do sistema, o tempo de simulação da simulação $CLOCK = t$ é atualizado para tempo de simulação $CLOCK = t_1$, e o aviso do evento executado é removido da FEL.
- A execução do evento iminente significa que um novo instantâneo do sistema para o tempo t_1 é criado. No momento t_1 , novos eventos futuros podem ou não ser gerados. Em caso positivo, novos eventos são agendados e colocados em sua posição correta na FEL.
- Este processo se repete até que a simulação termine.

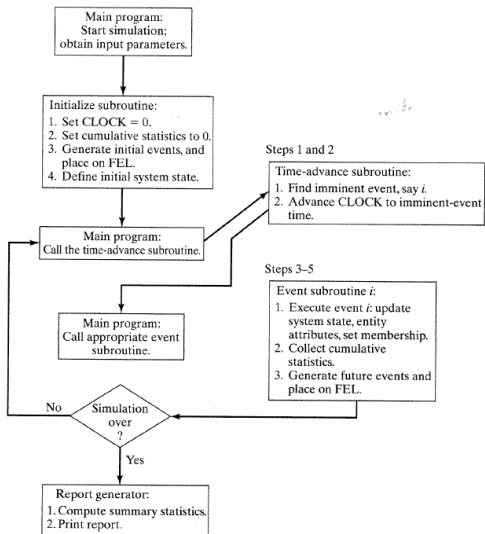
Tempo limite de simulação

- Determinado antes do início da simulação. Por exemplo: Tempo máximo de simulação de 100 minutos.
- A simulação termina quando um fato ocorre, por exemplo, executar a simulação até que o centésimo cliente seja atendido. Não pode ser determinado previamente.

Agendamento de eventos: Algoritmo

- Etapa 1. Remova o "aviso de evento" para o evento iminente da FEL.
- Etapa 2. Avançar o CLOCK para o horário do evento iminente.
- Etapa 3. Executar evento iminente: atualizar o estado do sistema.
- Etapa 4. Gere eventos futuros (se necessário) e coloque seu evento avisos na FEL de acordo com tempo do evento.
- Etapa 5. Atualizar estatísticas e contadores cumulativos.
- Etapa 6. Verifica se o tempo limite foi atingido.
- Etapa 7. Contabilizar as estatísticas.

Agendamento de eventos: Fluxograma



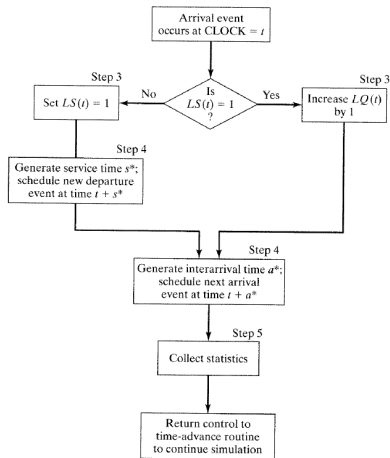
Exemplo

Uma pequena mercearia tem apenas um balcão de caixa. Os clientes chegam a este balcão em horários aleatórios com distribuição uniforme entre 1 a 8 minutos de intervalo entre as chegadas. Os tempos de serviço variam de acordo com uma distribuição uniforme de 1 a 6 minutos. Faça uma análise do sistema, simulando a chegada e o atendimento de clientes.

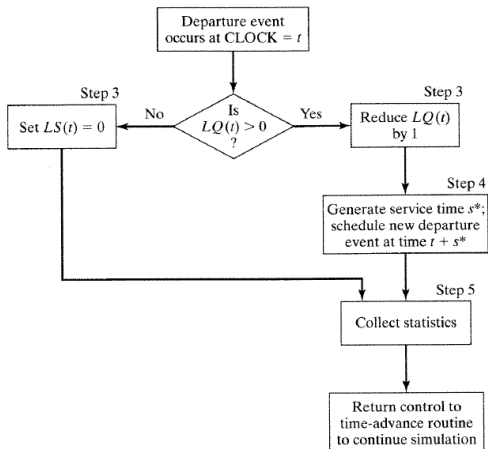
Componentes

- **Estado do sistema**
 - $LQ(t)$: Número de clientes esperando na fila
 - $LS(t)$: Número de clientes sendo servidos (0 ou 1) no tempo t .
- **Entidades** O caixa e o cliente.
- **Eventos** Chegada (A), partida (D), evento de parada ($E = 60$ minutos)
- **Avisos de eventos** (A, t), (D, t), (E, 60)
- **Atividades** Tempo entre chegadas e tempo de serviço
- **Atraso** Tempo do cliente gasto no tempo de espera

Exemplo: Fluxograma do evento de chegada



Exemplo: Fluxograma do evento de partida



- Tempo entre chegadas: 1,1,6,3,7,5,2,4,1,...
- Tempo de serviço: 4,2,5,4,1,5,4,1,4,...

Condições iniciais:

- No tempo 0, $LQ(0) = 0$, $LS(0) = 1$.

Estatísticas:

- **Utilização do caixa:** Tempo total do caixa ocupado (B) dividido pelo tempo de simulação (E);
- **Tamanho máximo da fila** (MQ)

Exemplo

- Tempo entre chegadas: 1,1,6,3,7,5,2,4,1,...
- Tempo de serviço: 4,2,5,4,1,5,4,1,4,...

CLOCK	System state		Future event list	Comment	Cumulative statistics	
	$LQ(t)$	$LS(t)$			B	MQ
0	0	1	(A, 1) (D, 4) (E, 60)	First A occurs ($a^* = 1$) Schedule next A ($s^* = 4$) Schedule first D	0	0
1	1	1	(A, 2) (D, 4) (E, 60)	Second A occurs: (A, 1) ($a^* = 1$) Schedule next A (Customer delayed)	1	1
2	2	1	(D, 4) (A, 8) (E, 60)	Third A occurs: (A, 2) ($a^* = 6$) Schedule next A (Two customers delayed)	2	2
4	1	1	(D, 6) (A, 8) (E, 60)	First D occurs: (D, 4) ($s^* = 2$) Schedule next D (Customer delayed)	4	2
6	0	1	(A, 8) (D, 11) (E, 60)	Second D occurs: (D, 6) ($s^* = 5$) Schedule next D	6	2

Exemplo

- Tempo entre chegadas: 1,1,6,3,7,5,2,4,1,...
- Tempo de serviço: 4,2,5,4,1,5,4,1,4,...

8	1	1	(D, 11) (A, 11) (E, 60)	Fourth A occurs: (A, 8) ($a^* = 3$) Schedule next A (Customer delayed)	8	2
11	1	1	(D, 15) (A, 18) (E, 60)	Fifth A occurs: (A, 11) ($a^* = 7$) Schedule next A Third D occurs: (D, 11) ($s^* = 4$) Schedule next D (Customer delayed)	11	2
15	0	1	(D, 16) (A, 18) (E, 60)	Fourth D occurs: (D, 15) ($s^* = 1$) Schedule next D	15	2
16	0	0	(A, 18) (E, 60)	Fifth D occurs: (D, 16)	16	2
18	0	1	(D, 23) (A, 23) (E, 60)	Sixth A occurs ($a^* = 5$) Schedule next A ($s^* = 5$) Schedule next D	16	2
23	0	1	(A, 25) (D, 27) (E, 60)	Seventh A occurs: (A, 23) ($a^* = 2$) Schedule next Arrival Sixth D occurs: (D, 23)	21	2

Diagrama de ciclo de atividade e Método das 3 fases

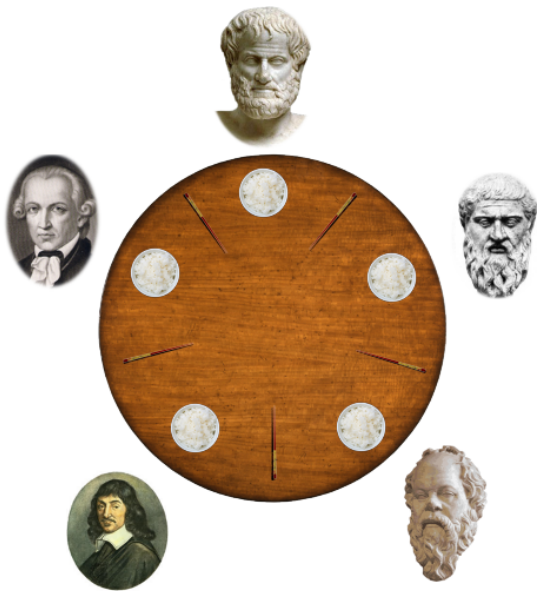
Diagrama de ciclo de atividade(ACD)

- O ACD é uma forma de modelagem das interações entre os objetos pertencentes a um sistema; particularmente útil em sistemas com fortes características de geração de filas.
- Os ACDs utilizam somente dois símbolos para descrever o ciclo de vida das entidades ou de objetos do sistema: uma circunferência, que representa uma “fila”, e um retângulo, que representa uma “atividade”.









Filósofos famintos

Um certo número de filósofos se sentam numa mesa circular. Entre cada par de filósofos há um garfo. Um filósofo só pode comer se ambos os garfos adjacentes a ele estiverem disponíveis; caso contrário, deverá aguardar. Após acabar de comer, o filósofo pensa por um certo tempo. Quando para de pensar, tentará comer novamente. Para comer, os filósofos levam um tempo uniformemente distribuído entre cinco e oito minutos, e, para pensar, um tempo obedecendo a uma distribuição normal com média de seis e desvio-padrão de um minuto. O objetivo do modelo é determinar a proporção de tempo que cada filósofo espera para comer, na média.

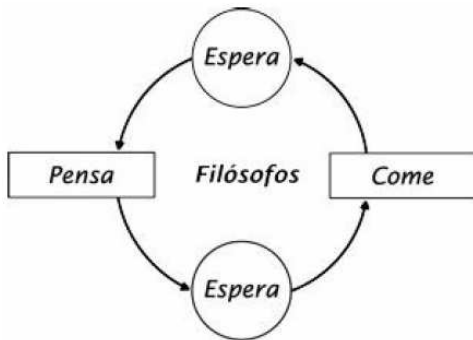


Entidades

- Filósofos
- Garfos.

Entidade	Estado	Símbolo
Garfo	É utilizado	
	Esperando	
	Comendo	
Filósofo	Esperando para pensar	
	Pensando	
	Esperando para comer	

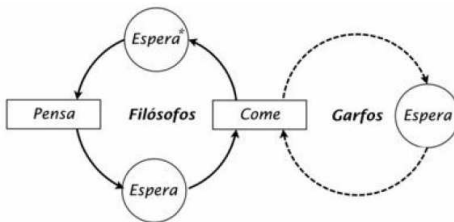
Ciclo de vida dos clientes



Ciclo de vida dos garfos













ACD completo



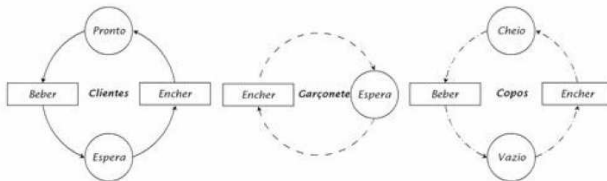
* Fila Fantasma

O cliente, a garçonete e o copo

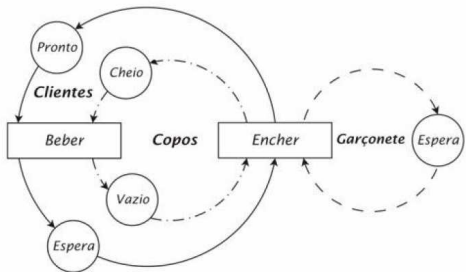
Considere uma versão simplificada de um pub, em que existem três entidades: “o cliente”, “a garçonete” e “o copo”. Quando o cliente entra no pub, ele pede uma cerveja, e a garçonete enche um copo para servi-lo. A seguir, o cliente bebe o conteúdo do copo. O cliente participa, portanto, das seguintes atividades: encher e beber. Tanto a garçonete quanto os copos participam da atividade “encher”.

Entidade	Estado	Símbolo
Cliente	Bebendo	
	Esperando	
	Encher	
	Pronto (para beber)	
Garçone	Enchendo	
	Esperando	
Copo	Sendo esvaziado	
	Vazio	
	Sendo enchido	
	Cheio	

Ciclo de vida dos clientes, garçoneiro e copos



ACD completo



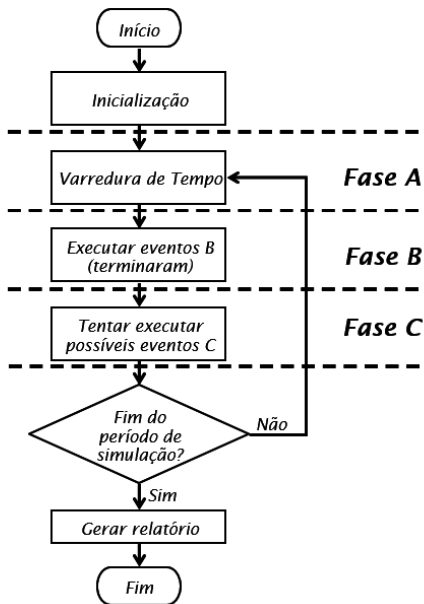
Simulação Manual

- Importante para o analista aumentar sua sensibilidade em relação a execução da simulação.
- Uma ferramenta de verificação se a lógica do modelo está coerente.
- Permite estabelecimento de prioridades onde existem.
- Promove um melhor entendimento do mecanismo de simulação

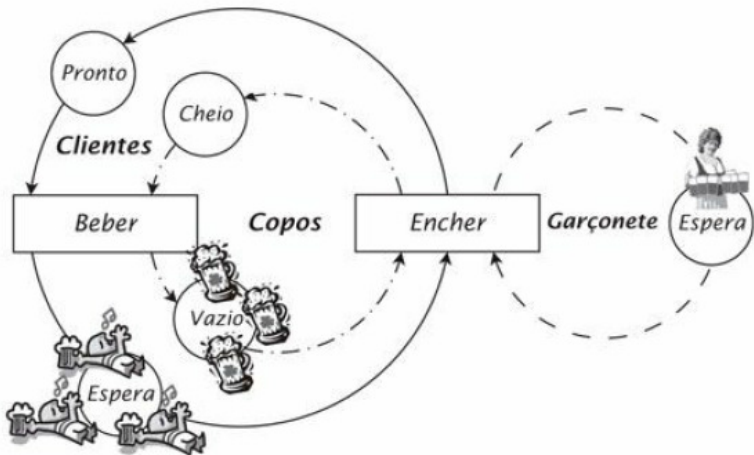
Método das três fases








- A: Checar o tempo de todas as atividades em progresso. Determinar a que ocorre primeiro. Avançar o relógio de simulação.
- B: Para as atividades que terminaram, mover as entidades para as respectivas filas.
- C: Procurar as atividades em uma ordem determinada e iniciar as que tem condição de começar. Mover as entidades das filas para a atividade. Amostrar e calcular o tempo de término da atividade.

Fluxograma

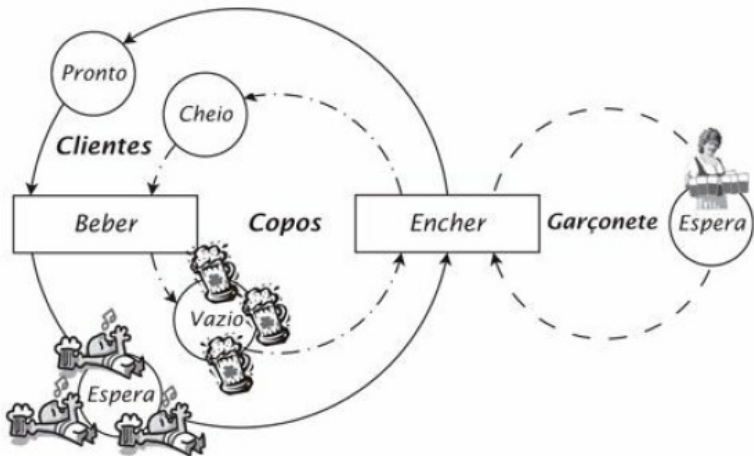


Para aplicar o método das três fases no exemplo do pub, vamos considerar que a atividade “beber” dura quatro minutos; encher dura três minutos; e que, no início da simulação, copos, clientes e garçonete estão dispostos como na Figura abaixo note que existem três copos, três clientes e uma garçonete.

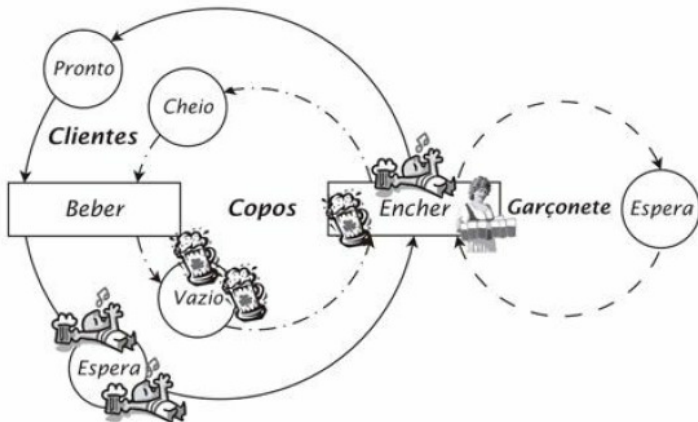


A	B	C
$T=0$ 	-0-	Encher começa e termina em 3
$T=3$ 	Encher termina	Beber começa e termina em 7 Encher começa e termina em 6
$T=6$ 	Encher termina	Beber começa e termina em 10 Encher começa e termina em 9
$T=7$ 	Beber termina	-0-
$T=9$ 	Encher termina	Beber começa e termina em 13 Encher começa e termina em 12
$T=10$ 	Beber termina	-0-
$T=12$ 	Encher termina	Beber começa e termina em 16 Encher começa e termina em 15
⋮	⋮	⋮

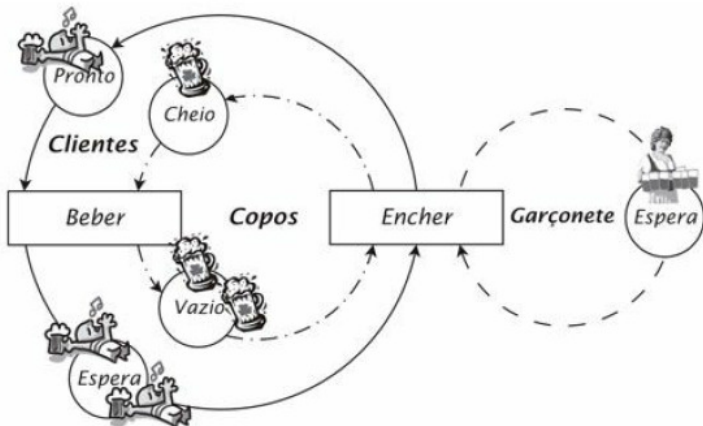
FASE A tempo 0



FASE C tempo 0



FASE B tempo 3



FASE C tempo 3

