

Diagrama de Estados

(ou Máquina de Estados ou Transição de Estados)

Material elaborado por: Profa. Ana Paula Lemke

Última atualização em 11/06/2018.

Diagrama de Estados

- É utilizado na modelagem comportamental de um **sistema completo**, de um **subsistema** ou **caso de uso** ou de um **objeto de uma classe**.
 - Melhor utilizado para a modelagem do comportamento de um objeto de uma classe, mostrando como o estado do objeto varia em decorrência de eventos.

Estado: condição ou situação existente na vida de um objeto.

Diagrama de Estados – Quando utilizar

- **Em princípio:** cada classe de um diagrama de classes possui um diagrama de estados correspondente.
 - **Na prática:** escolhe-se as classes com comportamentos complexos ou importantes dentro do contexto do sistema e derivam-se os diagramas de estados correspondentes.
 - Em outras palavras, usar somente para classes que realmente tenham diferentes estados durante os seus ciclos de vida.
-

Notação do Diagrama de Estados

Estado

- Um estado representa a situação de um objeto em um determinado momento, baseado no valor de um ou mais de seus atributos.
- Um estado pode ser subdividido em duas partes:
 - **Compartimento do nome**, que define o nome do estado.
 - **Compartimento de transições internas e ações do estado**, que mostra as ações ou atividades internas que são executadas enquanto o objeto se encontra neste estado.

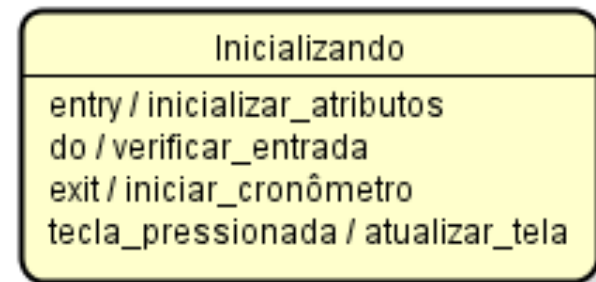


Estado

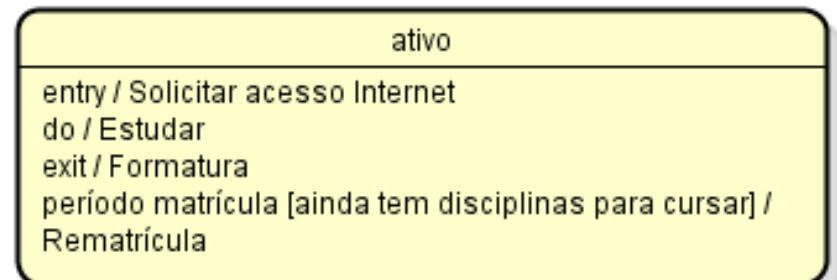
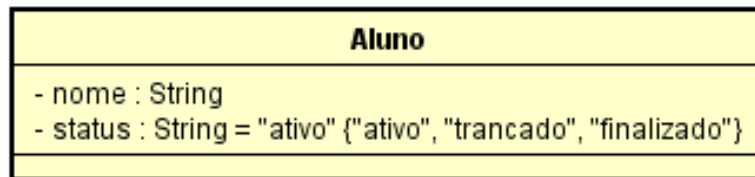
- Palavras reservadas do compartimento de transições internas:
 - **Entry**: ação realizada exatamente no momento em que se alcança o estado.
 - **Exit**: ação executada na saída do estado.
 - **Do**: permite especificar uma atividade não atômica (composta por mais de uma instrução) realizada no interior do estado. Depois que o objeto concluir eventuais ações de entrada, ele passará a executar a atividade indicada enquanto permanecer no estado.
 - Transições internas: lista de ações ou atividades internas ao estado que são executadas em resposta a eventos recebidos enquanto o objeto encontra-se no estado.
 - Quando há transições internas, as ações de entrada e saída não são novamente executadas, visto que não há uma mudança de estado propriamente dita.
-

Estado - Exemplos

- Estado “Inicializando” do atributo *status* de um Cronômetro



- Estado “Ativo” de uma instância de Aluno



Transições

- Os objetos mudam de estado a medida em que executem seus processamentos. A mudança/alteração do estado de um objeto é denominada **transição de estado**.
- Uma transição pode possuir um rótulo com a seguinte sintaxe:

Evento (**Argumentos**) [**Condição**] / **Ação**



Onde,

- ✓ **Evento**: indica o evento que ocasiona a transição de estados (evento de gatilho - *trigger*).
 - ✓ **Argumentos**: são as informações trazidas pelo evento.
 - ✓ **Condição**: é a condição que deve ser satisfeita para que a mudança de estado ocorra.
 - ✓ **Ação**: ação a ser realizada na mudança de estado.
-

Transições - Evento ou *trigger*

- **Evento** é a ocorrência de algum fenômeno que é reconhecido pelo objeto.
 - Exemplos de eventos em um software: clique de *mouse*, botão do teclado pressionado, mudança de ano, chamada de uma operação, recebimento de um e-mail, etc.
 - Um evento é considerado instantâneo.
 - Tipos de eventos:
 - Sinal: comunicação assíncrona entre objetos;
 - Chamadas: comunicação síncrona que representa a chamada de uma operação.
 - Tempo: representa a passagem de tempo.
-

Transições - Condição de guarda e Ação

- **Condição de guarda**
 - Expressão lógica avaliada quando a transição é lançada pelo evento de gatilho. Caso seja verificada, o objeto muda de estado. Caso contrário o objeto se mantém no estado corrente.
 - É possível haver diversas transições de estado a partir de um estado fonte para um mesmo evento de gatilho.
 - Caso nenhuma das transições que possam ser disparadas a partir da recepção de um mesmo evento de gatilho tenham suas condições de guarda satisfeitas, o evento é perdido (lembrar que o evento é instantâneo).
 - **Ação**
 - Executada imediatamente quando ocorre a transição. Em um sistema computacional, normalmente é a chamada de um método.
-

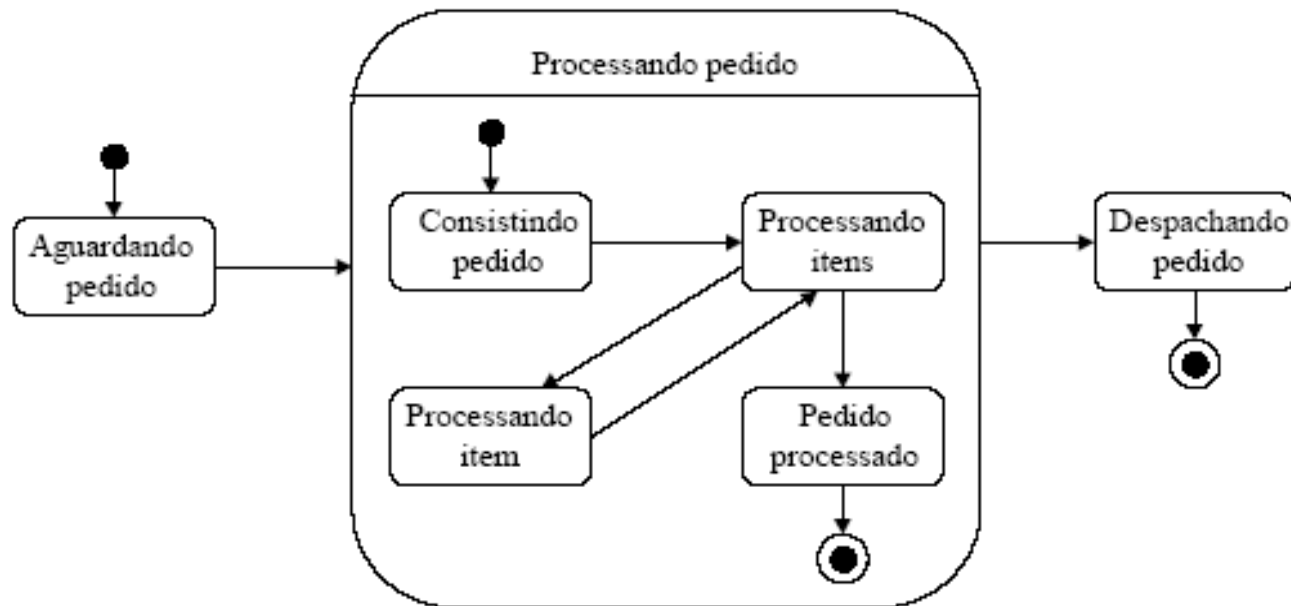
Estados inicial e final

- **Estado inicial:** indica o primeiro estado do objeto (também chamado de estado de partida do objeto).
 - Notação: ●
- **Estado final:** indica o último estado do objeto (momento de sua destruição ou desalocação).
 - Notação: ○

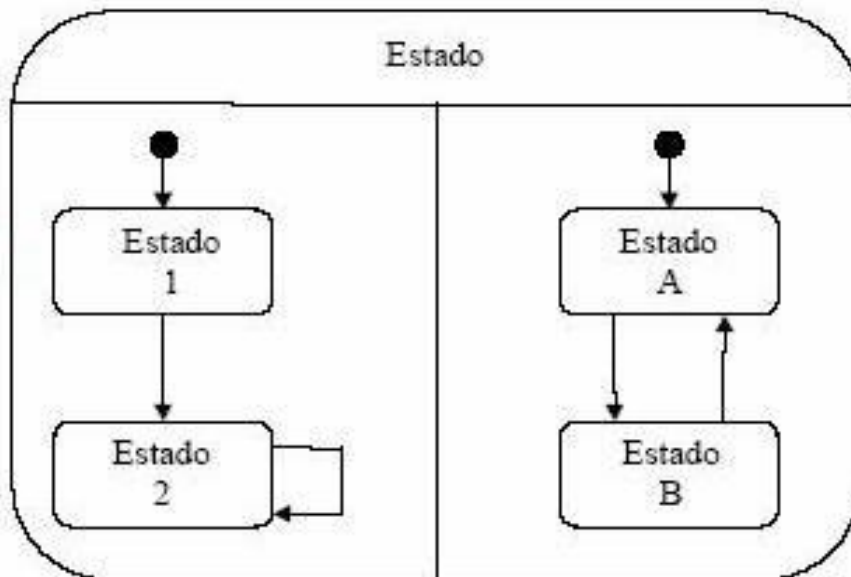
Sub-estados

- Em um diagrama de estados, um estado pode ser detalhado em sub-estados.
 - Um estado que possui sub-estados é chamado de **estado composto**.
 - Um estado composto pode conter:
 - Sub-estados sequenciais.
 - Sub-estados concorrente: é possível que um objeto esteja em dois (ou mais) sub-estados ao mesmo tempo, neste caso utiliza-se a concorrência.
-

Sub-estados sequenciais - Exemplo

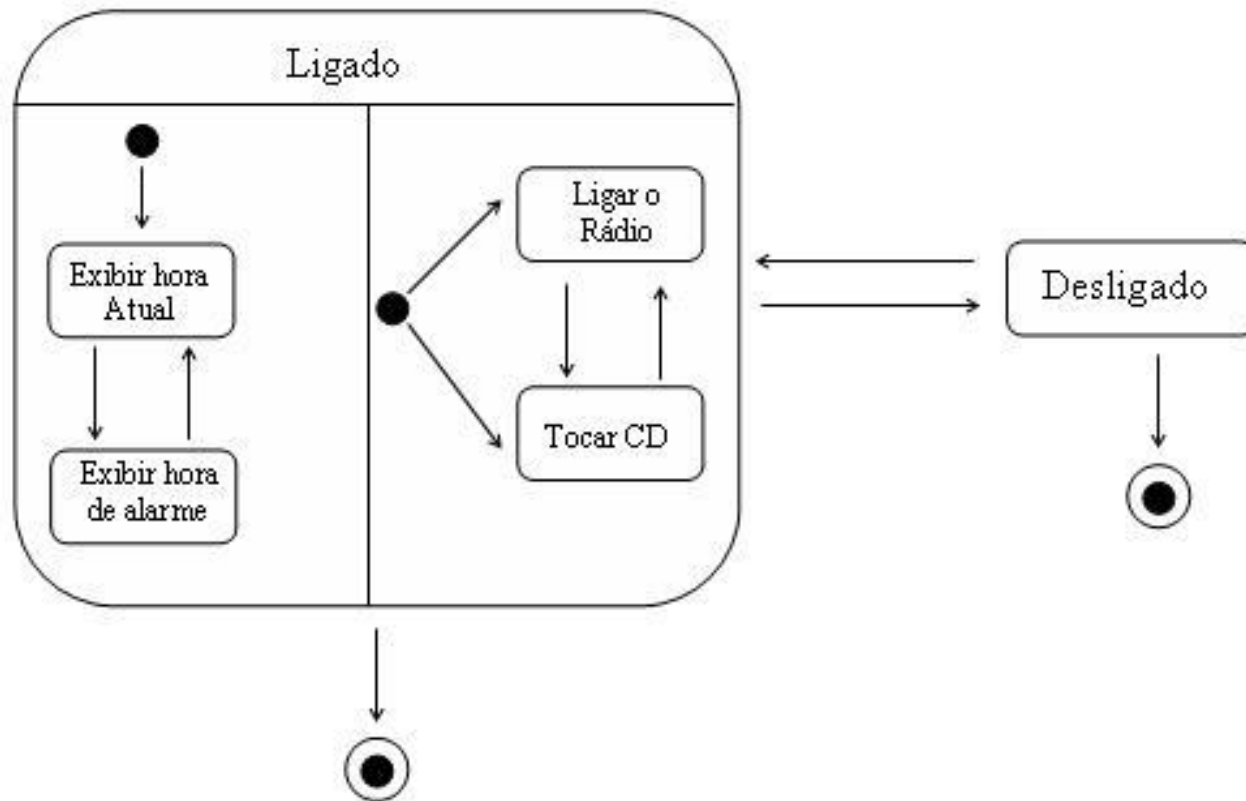


Sub-estados concorrentes - Exemplo



Um objeto pode estar simultaneamente nos sub-estados 1 e B, por exemplo. Entretanto deve-se garantir que os sub-estados 1 e 2 não dependam de A nem de B e vice-versa.

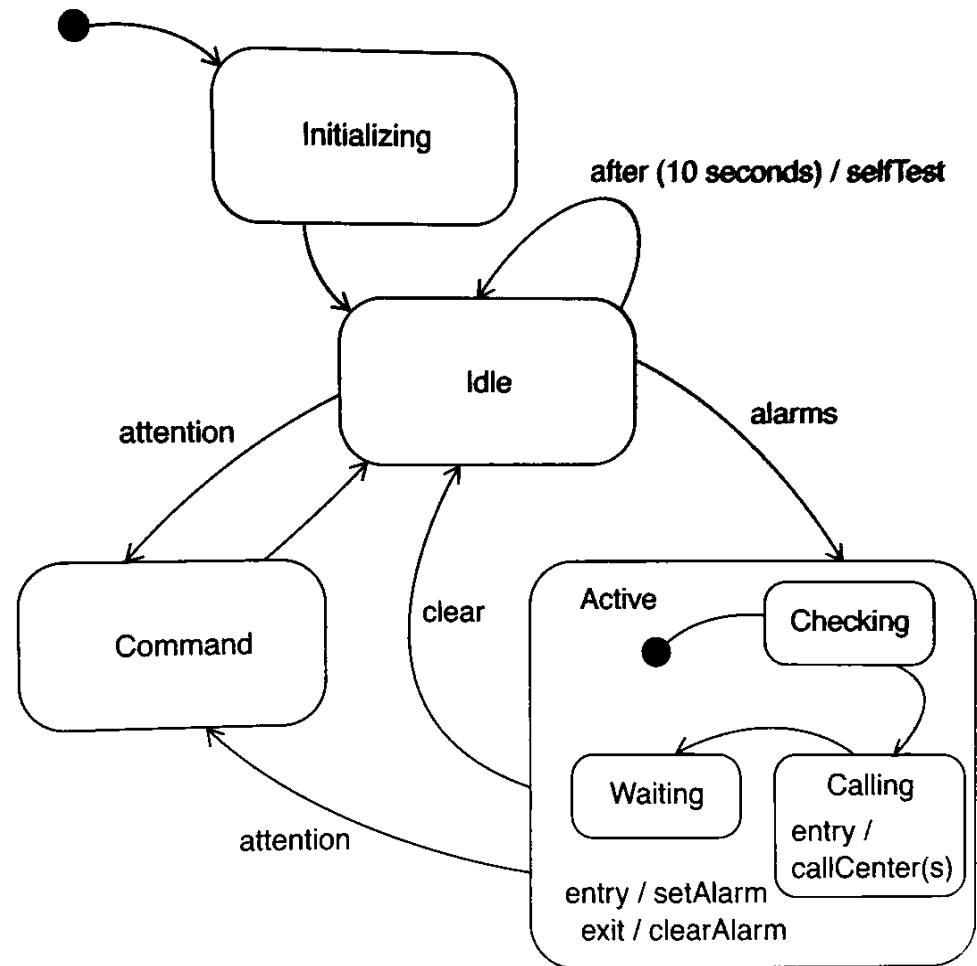
Sub-estados concorrentes - Exemplo



Exemplos de Diagramas de Estados

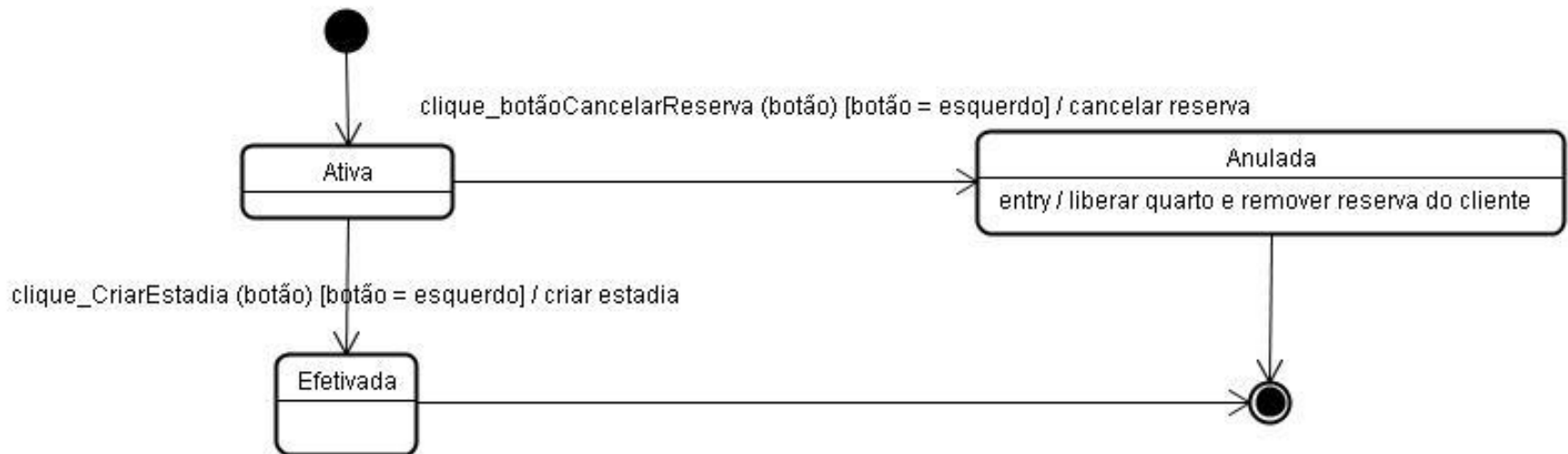
Diagrama de estados para o controlador de um **sistema de segurança doméstico**, responsável pela monitoração de vários sensores ao redor do perímetro da casa. A classe controlador tem 4 estados principais:

- **Initializing**: quando o controlador está inicializando;
- **Idle**: quando o controlador está pronto e aguardando alarmes ou comando do usuário.
- **Command**: quando o controlador está processando comandos do usuário;
- **Active**: quando o controlador está processando uma condição de alarme.



Exemplos de Diagramas de Estados

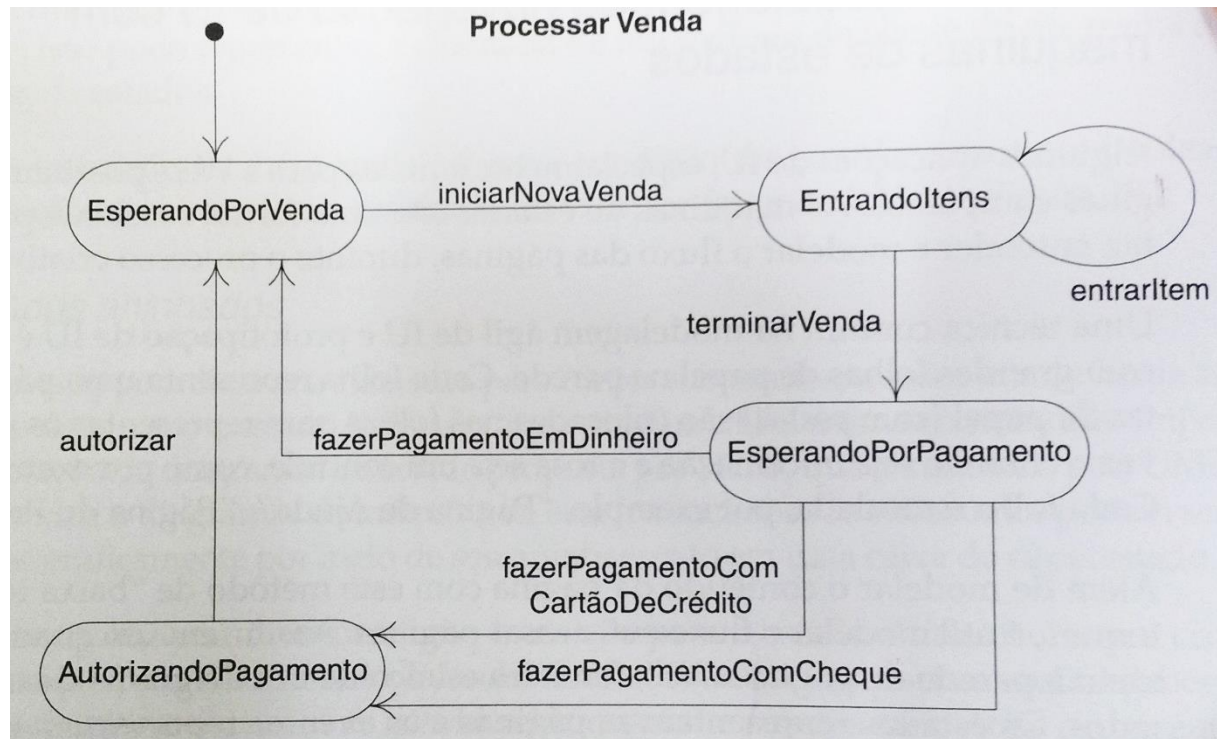
Diagrama de estados para um objeto da classe Reserva de um sistema de controle de ocupação de um Hotel.



Fonte: própria autora.

Exemplos de Diagramas de Estados

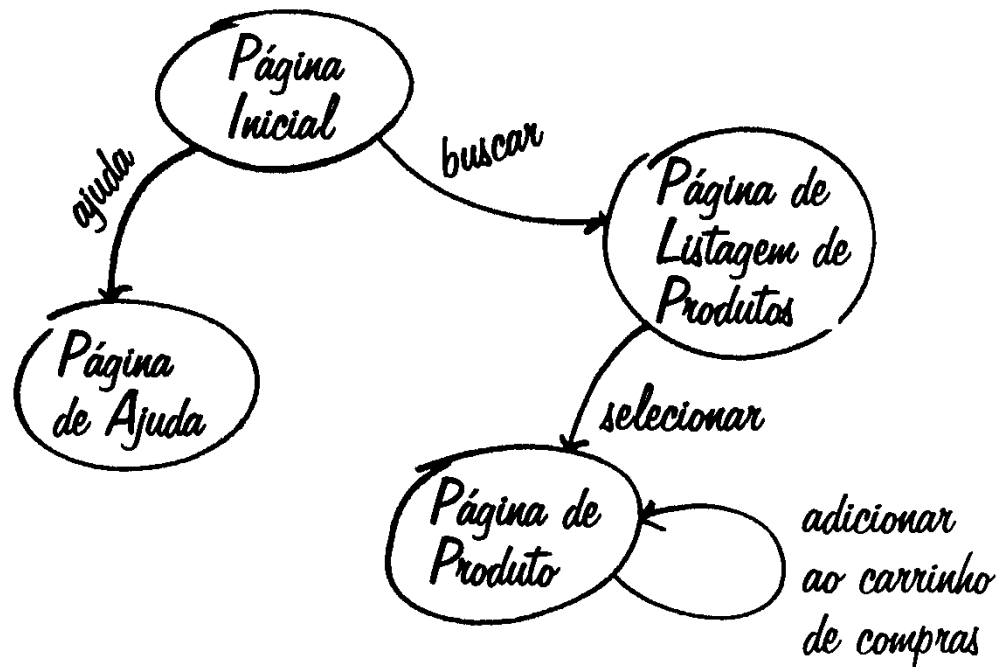
Diagrama de estados para sequência legal de operações em um caso de uso (**modelagem de um subsistema**)



Fonte: LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. (página 496).

Exemplos de Diagramas de Estados

Aplicação de máquina de estados na modelagem de navegação de páginas Web (**modelagem do sistema**)



Fonte: LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. (**página 496**).

Técnicas de Modelagem

- Quando **não há muitos estados** possíveis: estabelecer todos os estados possíveis de um objeto durante seu ciclo de vida, e depois conectá-los através de transições
 - Quando **há diversos estados** possíveis: modelar o ciclo de vida “normal” de um objeto (desde a sua criação até a sua destruição) e depois derivar todos os estados extraordinários, secundários ou de exceção.
-

Exercício 1

Crie um diagrama de estados para representar os estados possíveis que uma pessoa passa durante sua vida em relação ao seu **estado civil** (solteiro, casado, viúvo ou divorciado).

Exercício 2

Na modelagem de um sistema de Controle Acadêmico, há uma classe **Turma**, que apresenta os seguintes estados:

Aberta	A turma está aberta para receber inscrições de alunos, até a sua quantidade máxima. O professor, as salas e os horários foram alocados.
Lotada	A turma alcançou sua capacidade máxima em relação à quantidade de alunos inscritos.
Fechada	A turma está totalmente definida (os alunos estão inscritos).
Cancelada	A turma é cancelada. O evento de cancelamento pode acontecer a qualquer momento do ciclo de vida de uma turma.

Os eventos relevantes para objetos da classe **Turma** são a inscrição de um aluno, assim como as operações que realizam abertura, cancelamento e fechamento de uma turma. Além disso, o atributo que armazena a quantidade atual de alunos inscritos deve ser definido para monitorar quando esta quantidade atingir o valor máximo (considera-se que a quantidade máxima de alunos é uma regra de negócio definida no sistema).

Com base no descrito, faça o diagrama de máquina de estados para objetos da classe **Turma**.

Referências Adicionais

- Chonoles, M. and Schardt J. **“UML 2.0 for Dummies”**, 2003.
- Hamilton, Kim; Miles, Russell **“Learning UML 2.0”**, O'Reilly, 2006, 286 p.