

Unified Modeling Language (UML)

Universidade Federal do Maranhão – UFMA
Pós Graduação de Engenharia de Eletricidade
Grupo de Computação
Assunto: Diagrama de Estado

Autoria: Aristófanês Corrêa Silva

Adaptação: Alexandre César M de Oliveira

5 Diagrama de Estado

5.1 Definição

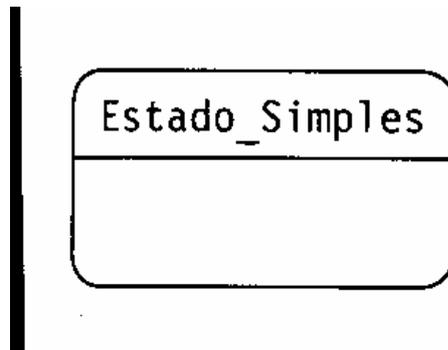
- Trata-se de um complemento para a descrição das classes, documentando os estados possíveis que objetos de uma certa classe podem assumir, além de mostrar ainda os eventos do sistema que geram tais mudanças.
- Os diagramas de estado não são escritos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos e onde o comportamento das classes é afetado e modificado pelos diferentes estados.
- Através da análise da mudança de estados dos tipos de objetos de um sistema, podemos prever todos os possíveis comportamentos de um objeto de acordo com os eventos que o mesmo possa sofrer.

- Diagramas de estado capturam o ciclo de vida dos objetos, subsistemas e sistemas.
- Eles mostram os estados que um objeto pode possuir e como os eventos (mensagens recebidas, timer, erros, e condições sendo satisfeitas) afetam estes estados ao passar do tempo.
- Todos os objetos possuem um estado que significa o resultado de atividades executadas pelo objeto, e é normalmente determinada pelos valores de seus atributos e ligações com outros objetos.
- Um objeto muda de estado quando acontece algo, o fato de acontecer alguma coisa com o objeto é chamado de evento.
- Os objetos de uma classe habitualmente possuem um ciclo de vida: são gerados, assumem posições durante a sua vida, dão origem a outros objetos em classes relacionadas e deixam de existir no momento de sua destruição
- Um diagrama de estado não capta – e não deve captar – todas as facetas e algoritmos possíveis da classe.

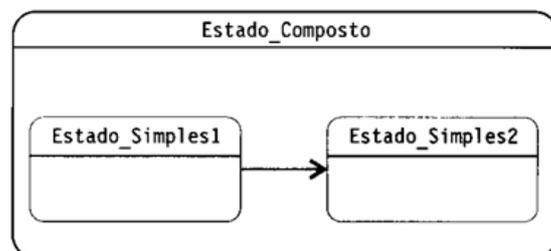
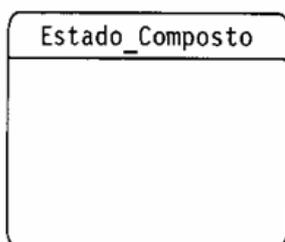
- Se o diagrama de estado está se tornando uma “miscelânea” de estados e condições, então muito provavelmente é necessário repensar sua classe.
- Devemos usar esse diagrama quando tivermos uma classe com mais de um atributo, que reflitam o estado de seus objetos em um determinado tempo, e que esses atributos mereçam ser modelados visando simplificar sua complexidade.
- Se o relacionamento de classes não está claro o suficiente em função do estado dos objetos, isso será uma pista de que deve usar este diagrama. Essa percepção é pessoal

5.2 Notações do diagrama

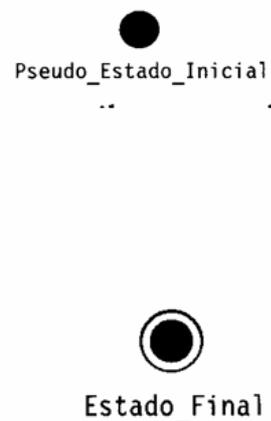
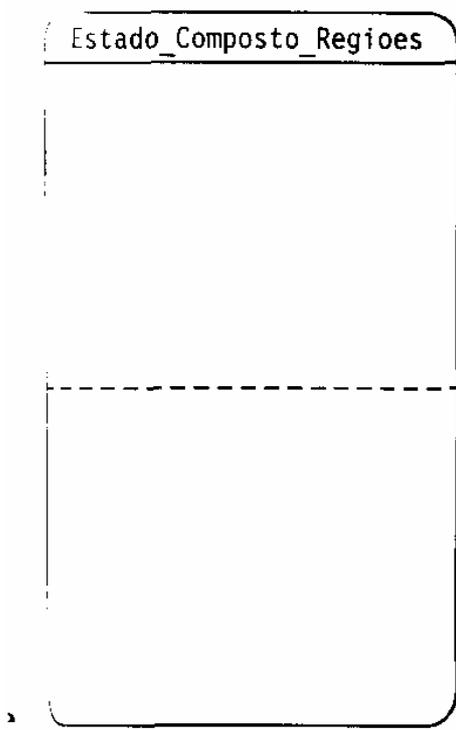
- Um retângulo designa um estado simples. O mais comum usado nos diagramas de estado. Seus cantos são arredondados. Um único valor, de um atributo de uma classe, pode ser representado por um estado simples.
- O estado tem 2 compartimentos, um com seu nome e o outro com suas ações internas.



- Um retângulo maior designa um estado composto ou máquina de estado, significando que nele existirão outros estados.

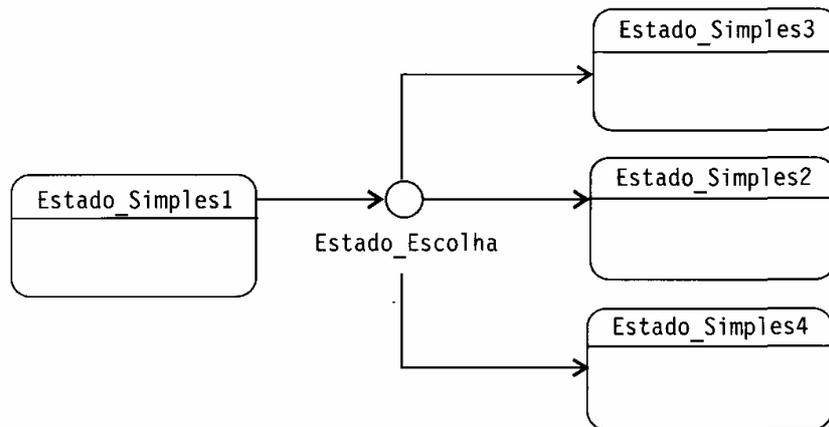


- Estado composto por regiões. Essas regiões são separadas por uma linha pontilhada, vertical ou horizontal, dentro do estado. Cada região pode ter um nome diferente e seus estados separados. Este estado é conhecido como estado concorrente.

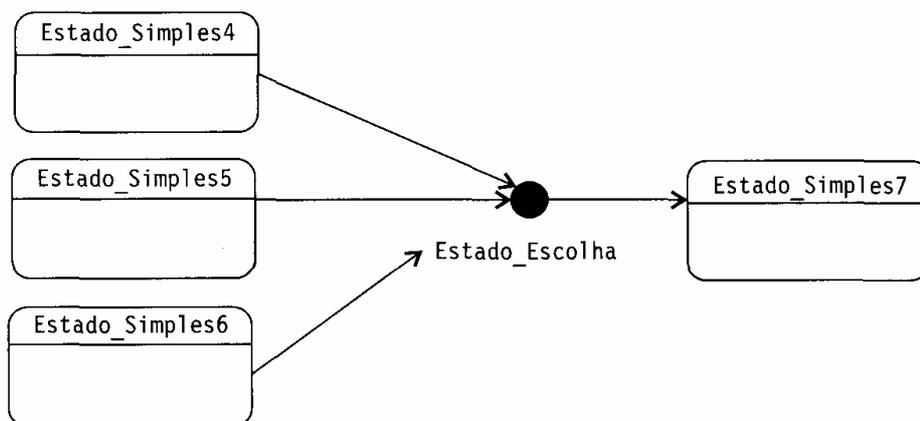


- Um círculo preenchido designa um estado inicial chamado de pseudo-estado inicial.
- O estado final é representado por um círculo vazado e, dentro deste, um círculo cheio. O último estado de um objeto será representado por este estado final.

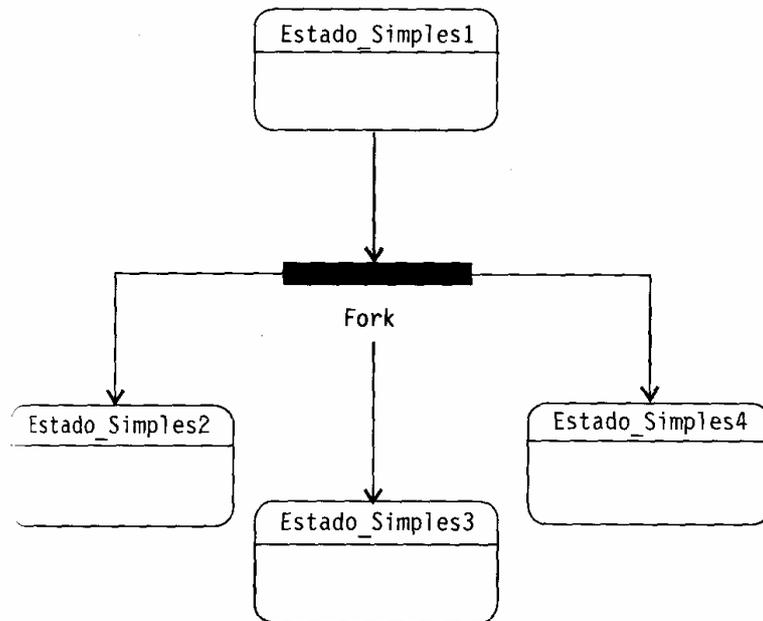
- Um círculo simples é um estado de escolha. Ao se chegar a um estado de escolha, o software deve tomar uma decisão e ir para qualquer estado possível, dos que saem do estado de escolha.



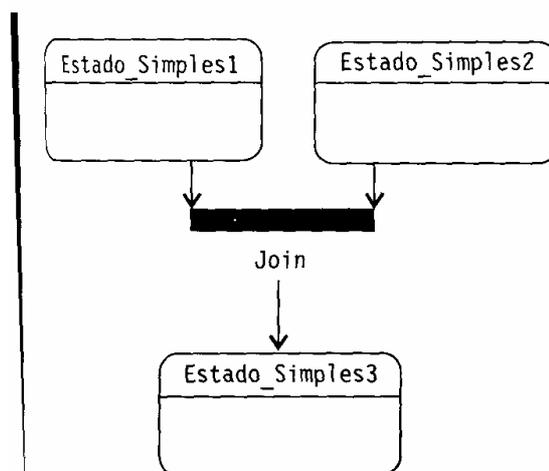
- Podemos ter a necessidade de representar que vários estados podem chegar em um único ponto. Este ponto é conhecido como junção: um círculo preenchido.



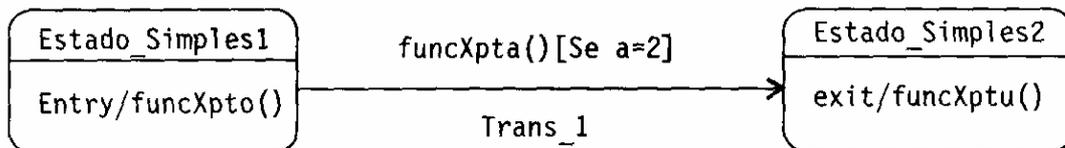
- A chegada de um estado a um *fork* pode desencadear um ou vários estados concorrentes.



- O estado *join* representa vários estados convertendo-se em apenas um.



- *Entry action* significa que a todo o momento que entrarmos neste estado esta operação será realizada, no caso, na operação *funcXpto()*

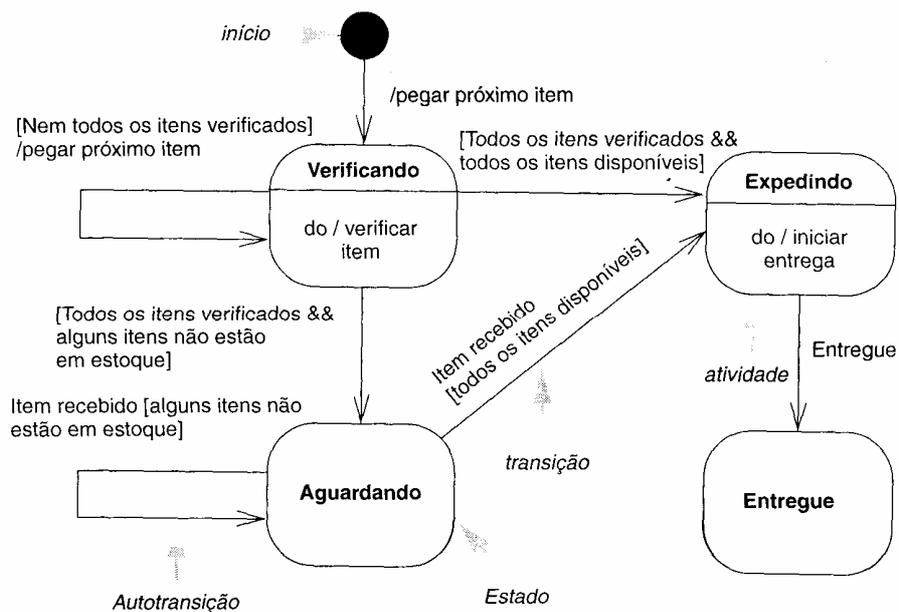


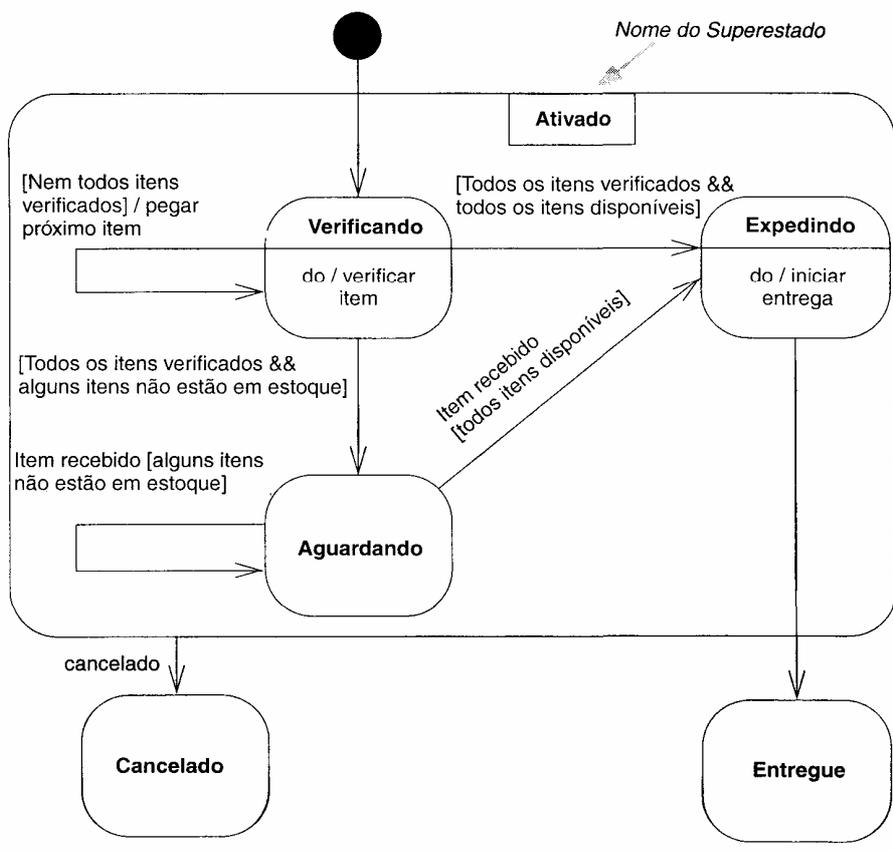
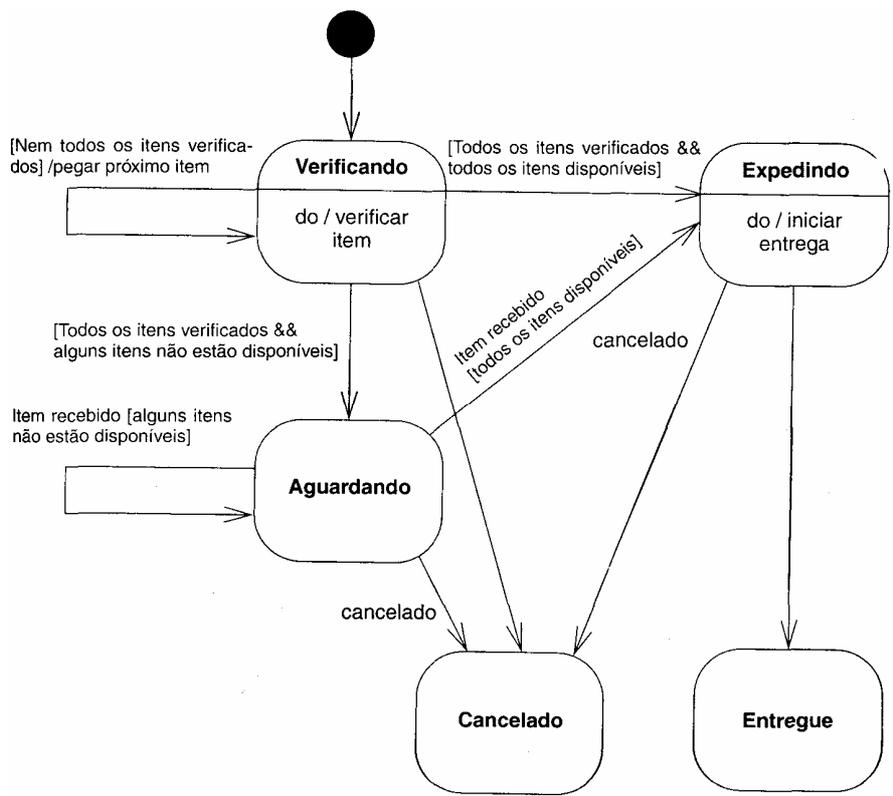
- A transição está indicada com a operação (*funXpta()*) e também com uma condição [*se a = 2*]. Esta condição é chamada de *Guard*
- No estado_*simples2* temos *exit action* significando que toda vez que sairmos do estado, esta operação será realizada, no caso *funcXptu()*.
- Temos também um tipo de ação que é executado durante todo o tempo de permanência naquele estado. Esta ação é representada por *do action*.
- Transição tem 3 partes opcionais: evento[guarda]/ação
- Guarda é uma condição lógica que devolve SIM ou NÃO
-

5.3 Dicas

- Ações são associadas com transição e são considerados como processos de curta duração, não podendo ser interrompidos
- Atividades são associados a estados e podem levar mais tempo. Uma atividade pode ser interrompida por algum evento.

5.4 Exemplos





Exemplo (diagrama de estados concorrentes)

