



Desenvolvimento de Sistemas Software

Aula Teórica 9

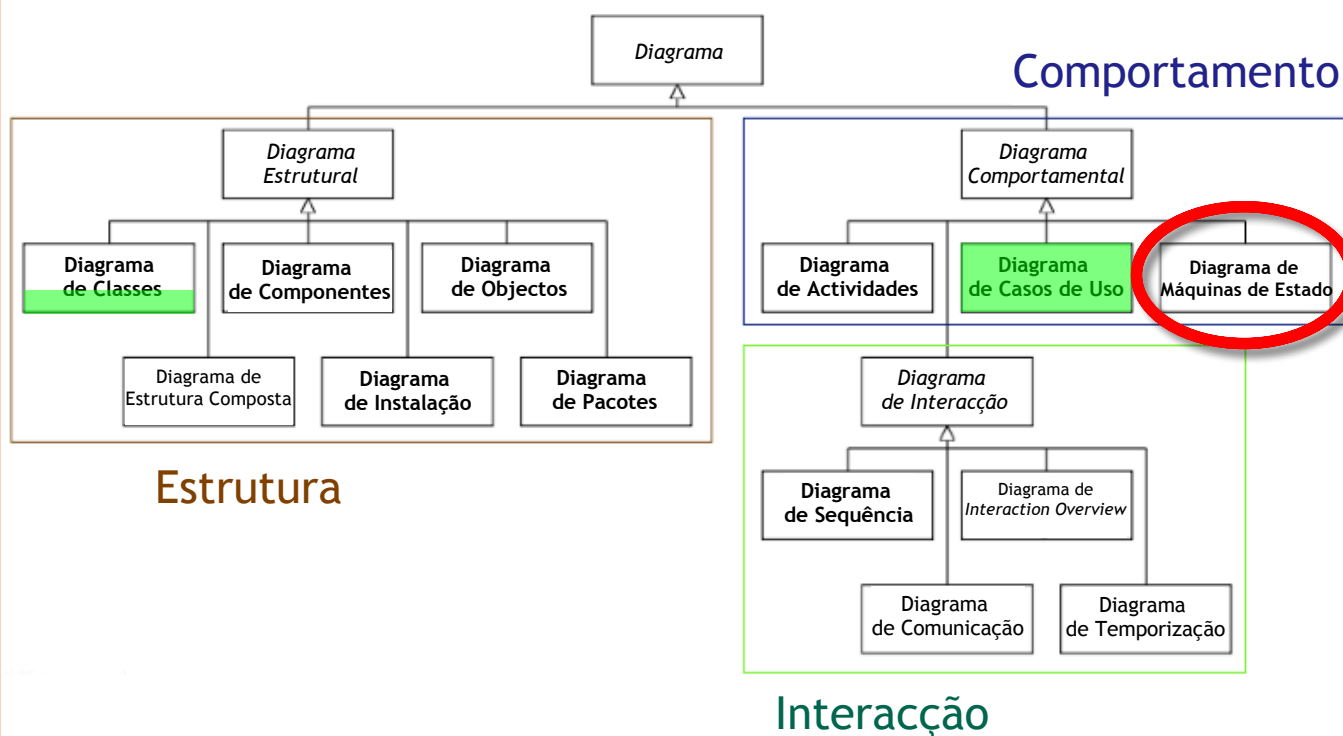
Modelação de comportamento / Máquinas de Estado

v. 2017/18

193



Diagramas da UML 2.x



v. 2017/18



Introdução aos Diagramas de Estado – Aplicação

- Os Diagramas de Estado permitem modelar o comportamento de um dado objecto/sistema de forma global.
- A ênfase é colocada no estado do objecto/sistema – modelam-se todos os estados possíveis que o objecto/sistema atravessa em resposta aos eventos que podem ocorrer.
- Úteis para modelar o comportamento de um objecto de forma transversal aos use case do sistema.
- Devem utilizar-se para entidades/classes em que se torne necessário compreender o comportamento do objecto de forma global ao sistema.
- Nem todas as entidades/classes vão necessitar de diagramas de estado.
- Úteis para modelar o comportamento do sistema como um todo (em particular a interface com o utilizador).

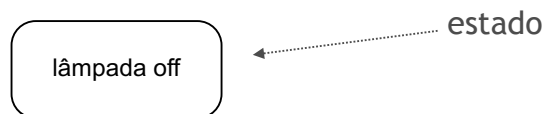
v. 2017/18



Diagramas de Estado

Notação base

- Estado – define uma possível estado do objecto (normalmente traduz-se em valores específicos dos seus atributos)



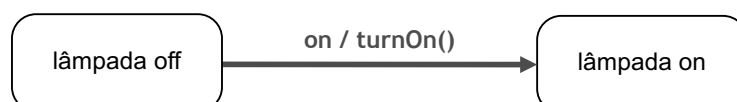
- Estado inicial – estado do objecto quando é criado



- Estado final – destruição do objecto



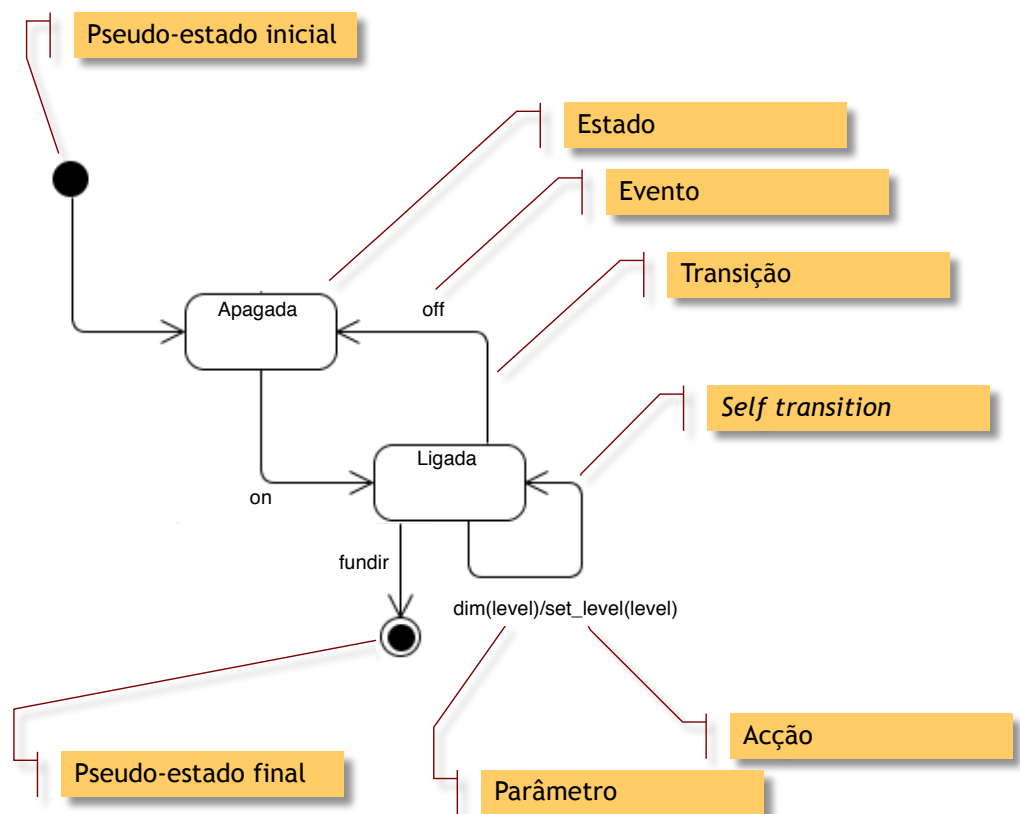
- Transições – evento[guarda]/acção (todos são opcionais!)



v. 2017/18



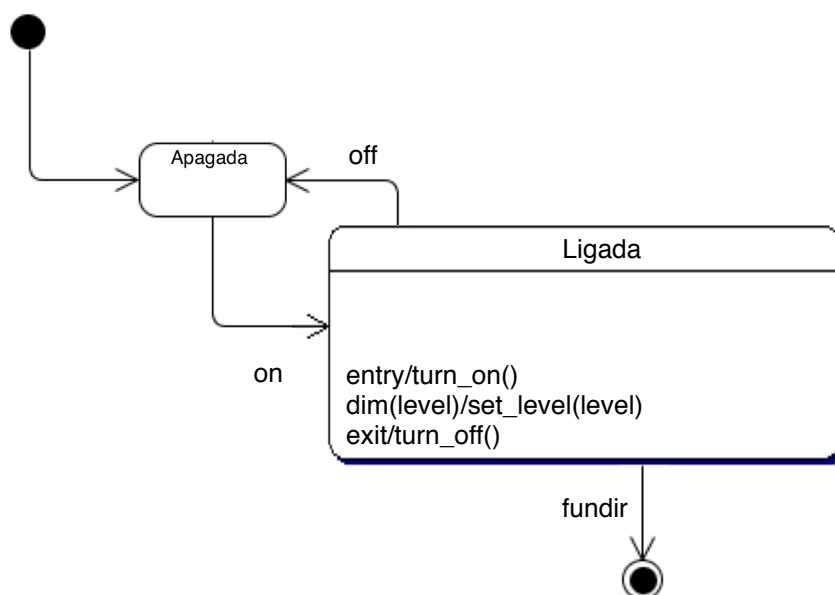
Maquina de Estados básica



v. 2017/18



Actividades internas



v. 2017/18



Actividades internas

- Actividades que não provocam transições de estado...

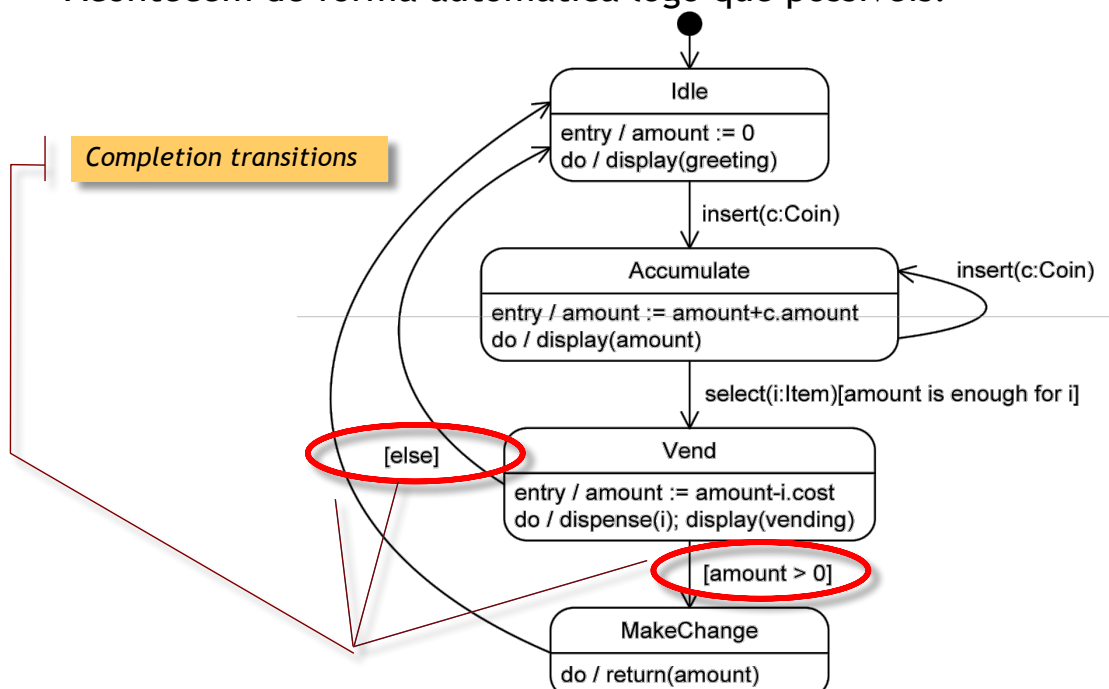
entry/acção	• “acção” é automaticamente executada quando o objecto entra no estado;
evento/acção	• “acção” é automaticamente executada se “evento” ocorrer (transição interna);
do/acção	• “acção” é continuamente executada enquanto o objecto estiver no estado;
evento/defer	• “evento” é deferido até o estado actual ser abandonado;
exit/acção	• “acção” é automaticamente executada quando o objecto sai do estado.

v. 2017/18



Completion transitions

- Sem eventos associados
- Acontecem de forma automática logo que possíveis.

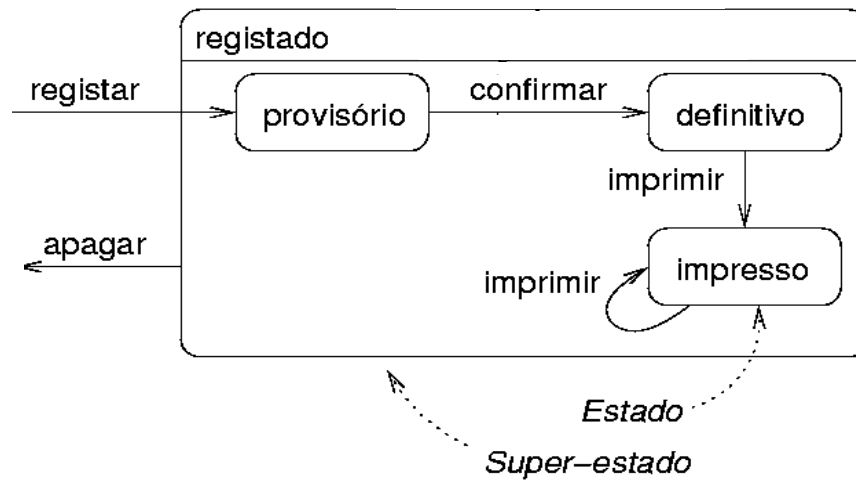


v. 2017/18



Estados e estados compostos (super-estados)

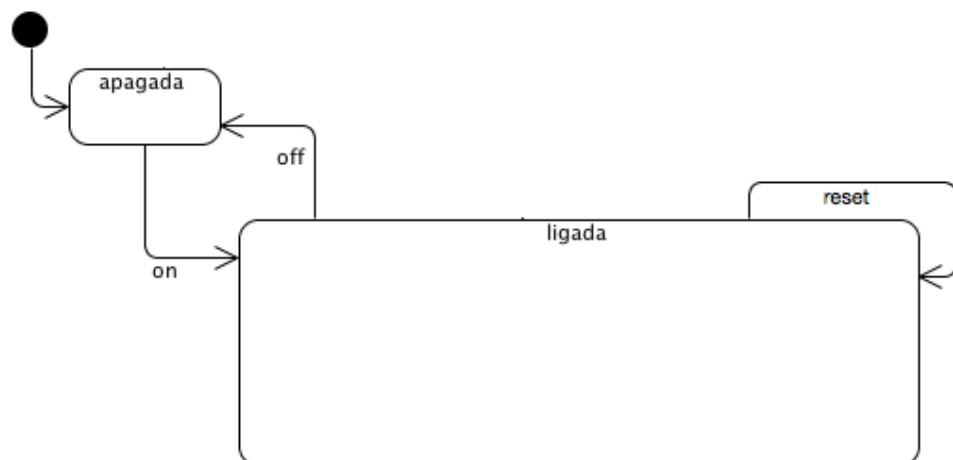
- Super-estados — permitem estruturar os modelos



v. 2017/18



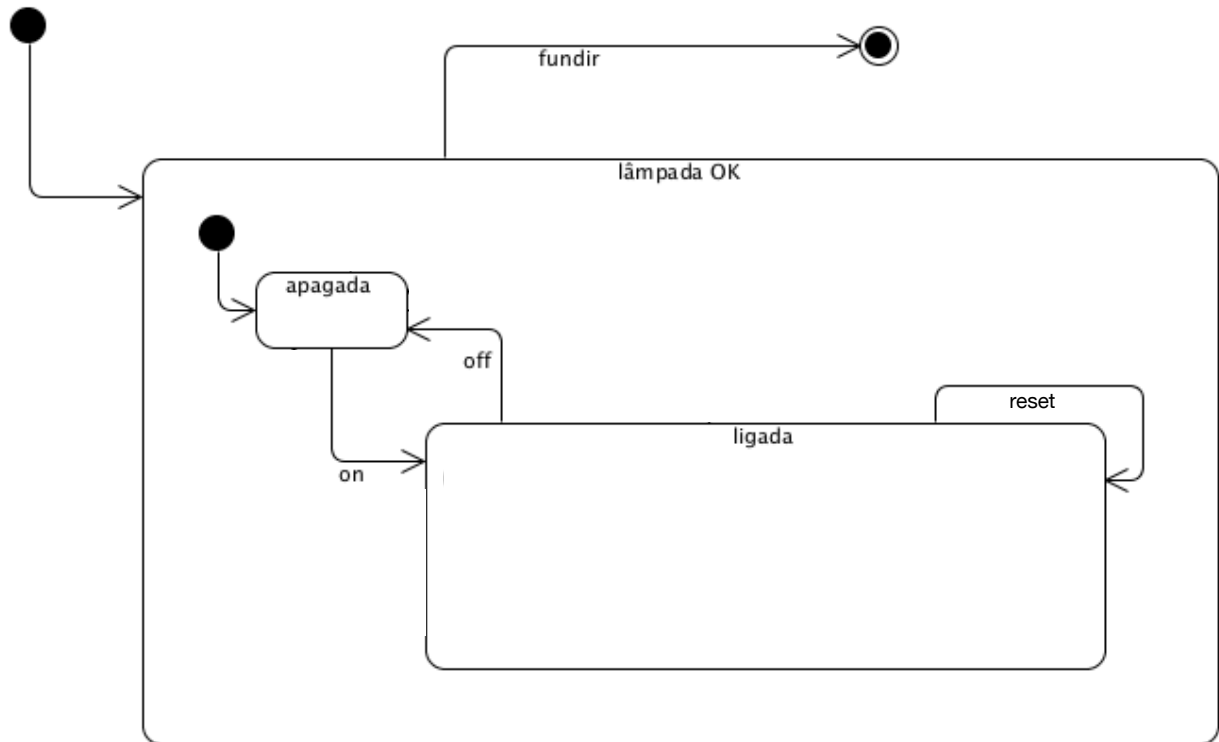
Estados e estados compostos (super-estados)



v. 2017/18



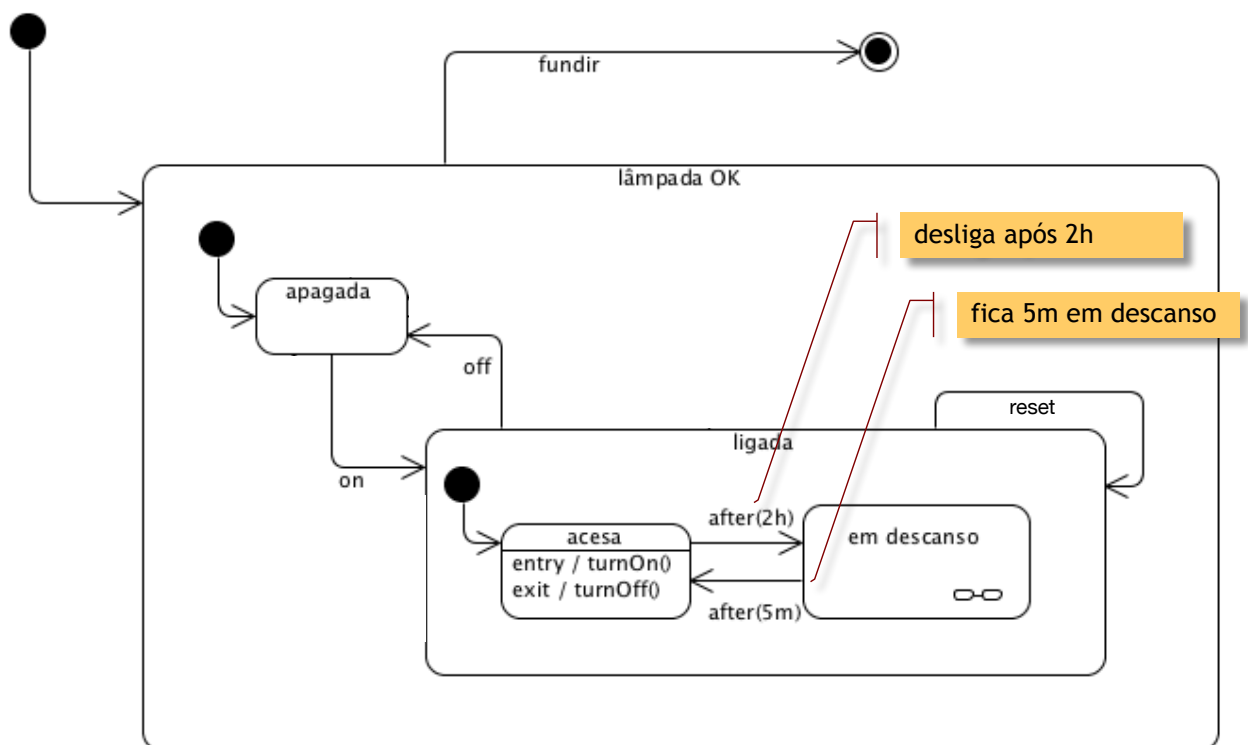
Estados e estados compostos (super-estados)



v. 2017/18



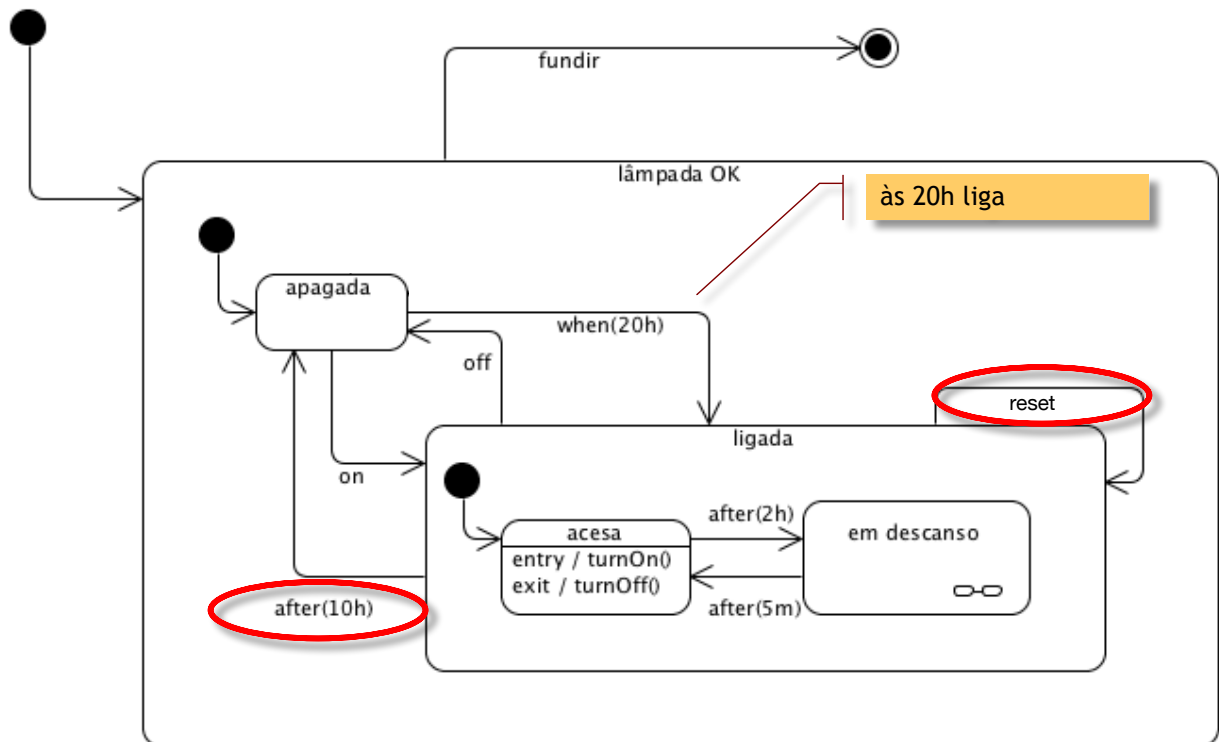
Eventos when/after



v. 2017/18



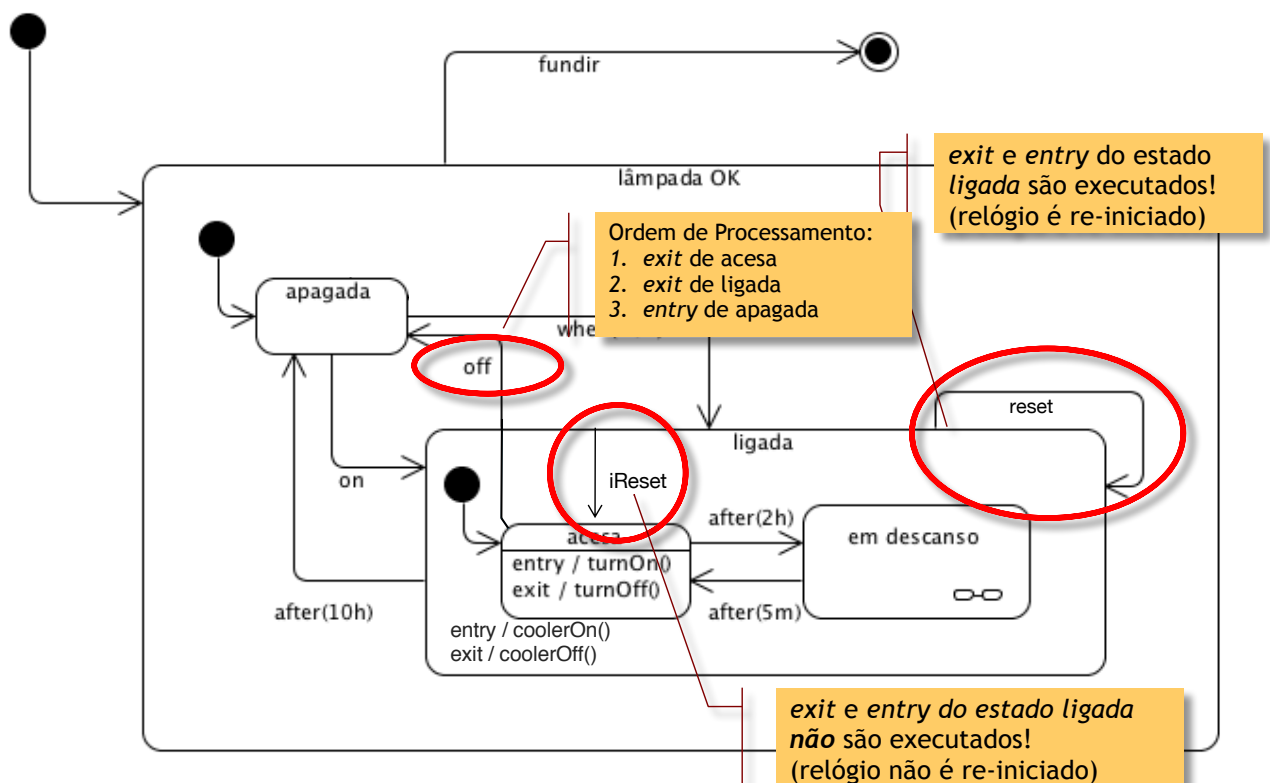
Eventos *when/after*



v. 2017/18



Transições vs. actividades internas

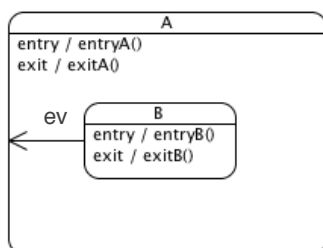


v. 2017/18



Transições locais vs. transições externas

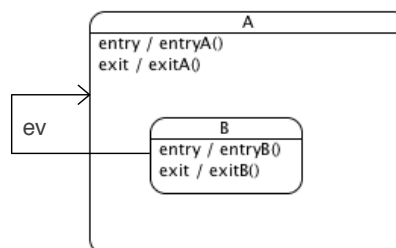
Transições locais



(sub-estado para super-estado)

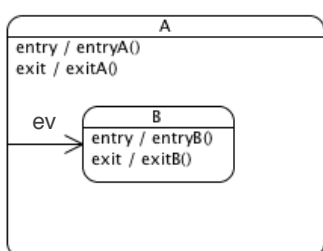
Executa:
1. exitB()

Transições externas



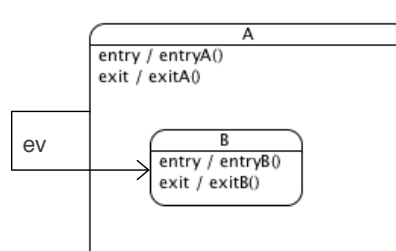
(sub-estado para super-estado)

Executa:
1. exitB()
2. exitA()
3. entryA()



(super-estado para sub-estado)

Executa:
1. entryB()



(super-estado para sub-estado)

Executa:
1. exitA()
2. entryA()
3. entryB()

v. 2017/18



Resumo da notação (até agora)

	Estado
	Estado composto
	Estado submáquina
	Pseudoestado inicial
	Estado final
	Transição (evento [condição] / acção)
	Transição para o próprio (evento [condição] / acção)
	Transição interna

v. 2017/18



Diagramas de Estado (*Statecharts*)

Sumário

- Introdução aos Diagramas de Máquinas de Estado — Aplicação
- Noções base: estados e transições
- Notação base
 - Estados
 - Pseudo-estado inicial
 - Pseudo-estado final
 - eventos *when/after*
 - Transições
- Actividades internas
- Estados simples vs estados Compostos