# Engenharia de Software

Modelos de Dinâmica – Parte 1

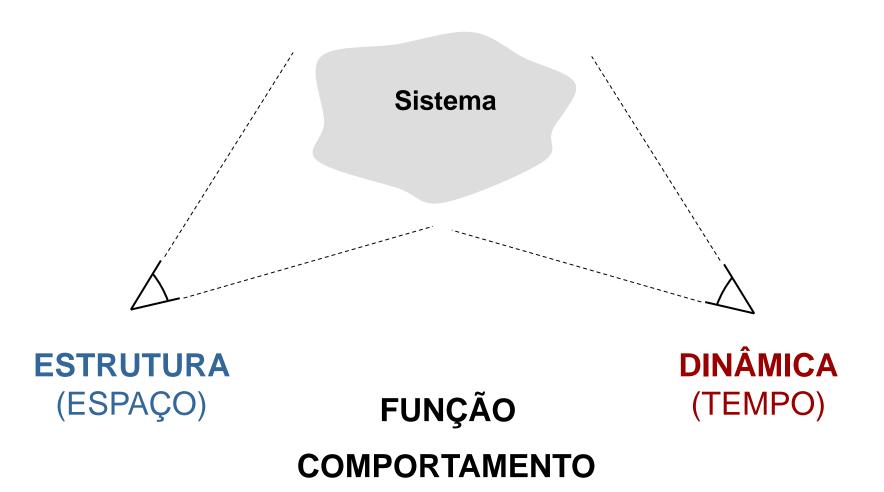
Luís Morgado

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

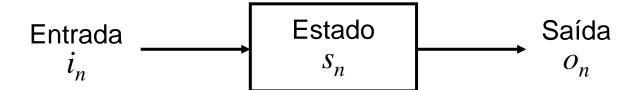
# Especificação de um Sistema

### Perspectivas de modelação



## Modelação da Dinâmica

- Arquitectura de um sistema
  - Organização no espaço (estática)
    - Estrutura
  - Organização no tempo
    - Dinâmica



### **Estrutura**

 Denota as partes e as relações entre partes de um sistema

### Dinâmica:

 Denota a forma como as partes e as relações entre partes de um sistema evoluem no tempo

### Comportamento

 Denota a forma como o sistema age ou reage perante os estímulos do ambiente envolvente

### Dinâmica de um Sistema

## Sistema computacional

- -Estado
  - Conjunto de dados armazenados aos quais o sistema computacional tem acesso num determinado instante no tempo
- A saída de um sistema computacional, a qualquer momento, é completamente determinada pelas suas entradas e pelo seu estado

## Dinâmica de um Sistema

### Estado

- Configuração da estrutura de um sistema discriminadora do comportamento do sistema
  - Representa situações relevantes da evolução de um sistema
  - Exemplo: quando se prime uma tecla de um teclado o código gerado depende do estado do CAPS-LOCK (ON / OFF)
  - Comportamento depende apenas de alguns aspectos da história de evolução, neste caso, não depende de quantas teclas foram pressionadas anteriormente

## Estado e Variáveis de Estado

### Estado no sentido lato (micro)

- Cada estado representa um conjunto distinto de configurações das partes do sistema (e.g. os valores da memória)
- Problema
  - Mesmo para programas pequenos o número de estados é impraticável
    - Exemplo: programa com uma variável do tipo inteiro 32 bits
      - » 2^32 = 4.294.967.296 estados!

### Estado como generalização (macro)

- Abstração
- Discriminação de padrões comportamentaos

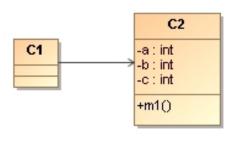
### Variáveis de estado

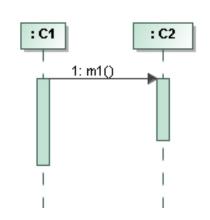
- Aspecto quantitativo
- Exemplo:
  - Parar ao fim de 1000 teclas pressionadas
  - 1000 estados ?
  - Variável auxiliar (e.g. contador de teclas)
- Podem ser úteis mas tornar a modelação menos compreensível e adaptável

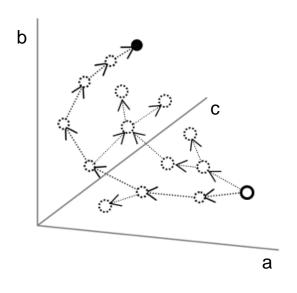
### **Estrutura**

### Comportamento

### Dinâmica

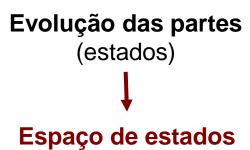


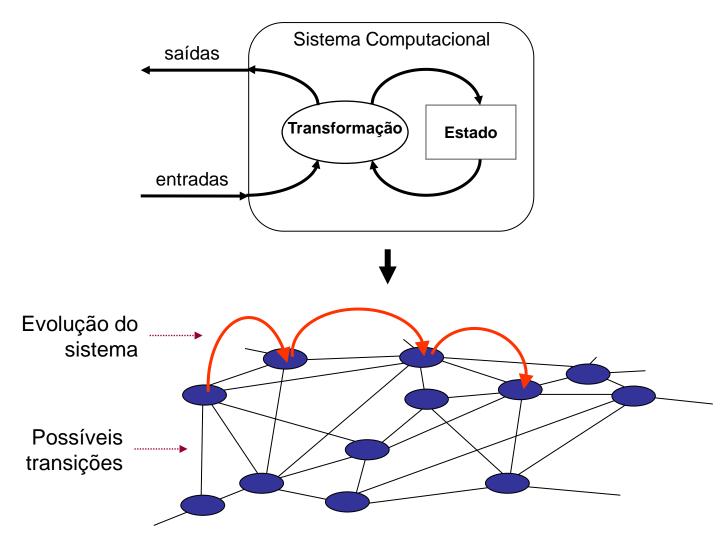




Tipos de partes (classes)

Interacção entre partes (instâncias)

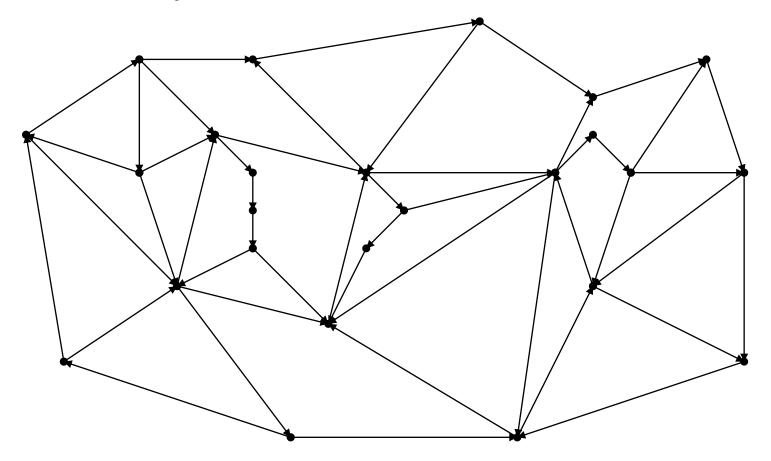




Espaço de estados

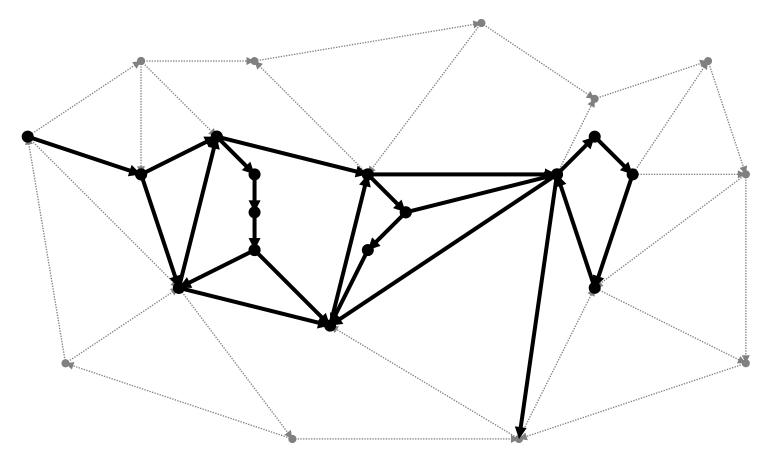
Representação de um espaço de estados sob a forma de um grafo

- Vértices Estados
- Arcos Transições de estado



Espaço de estados (grafo)

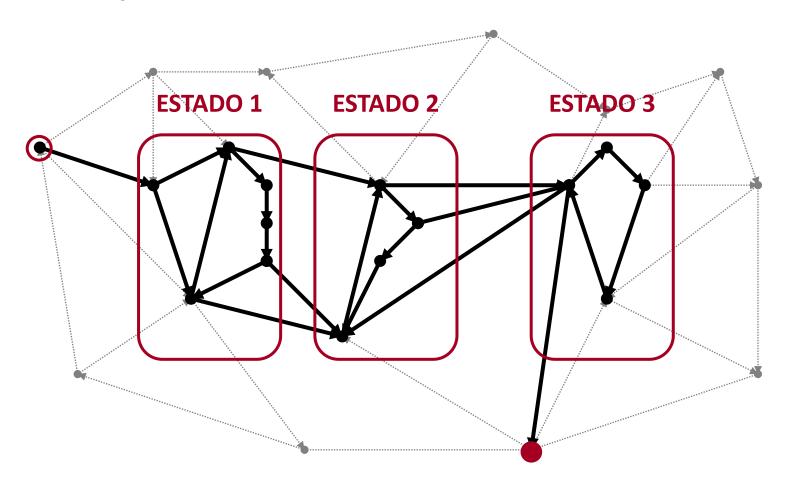
Definição de **comportamento válido** com base no espaço de estados



Apenas o comportamento especificado é admissível

- Abstrai complexidade procedimental
- Definição robusta de comportamento

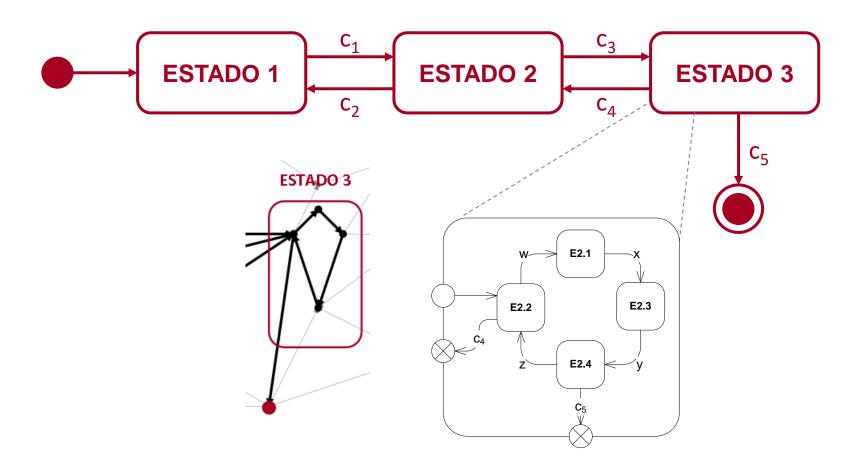
### **ABSTRACÇÃO**



Permite gerir a complexidade da dinâmica

- Modularização
- Detalhe selectivo

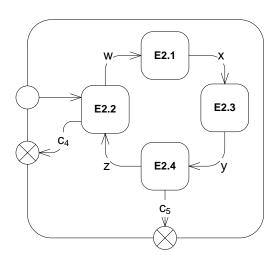
# Diagramas de Transição de Estado



ABSTRACÇÃO → CONTROLO DE COMPLEXIDADE

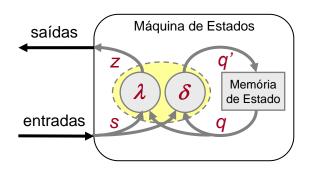
## Máquinas de Estados

### Diagrama de Transição de Estado



Especifica a dinâmica de um sistema ou parte de um sistema

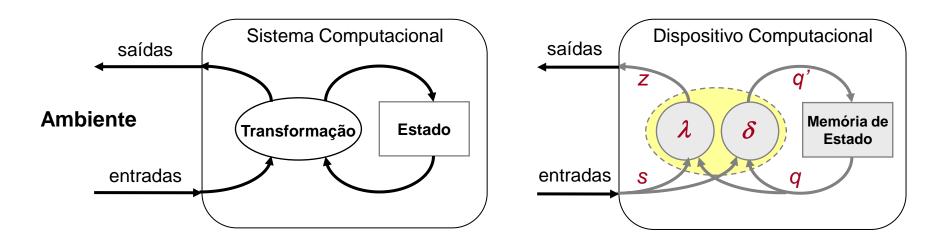
### Máquina de Estados



Concretiza a dinâmica de um sistema ou parte de um sistema

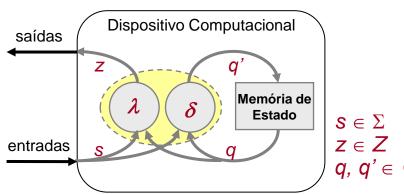
## Modelação da Dinâmica

A dinâmica pode ser expressa como uma função de *transformação* que, perante o estado actual e as entradas actuais, produz o estado seguinte e as saídas seguintes.



Esta caracterização de um sistema computacional é **independente da forma concreta como este possa ser implementado** em termos físicos. O suporte físico pode ser, por exemplo, mecânico, electrónico, biológico.

- Entradas e saídas abstraídas em termos dos conjuntos de símbolos que nelas podem ocorrer:
  - Esses conjuntos de símbolos são designados alfabetos.
  - Consideremos um alfabeto de entrada  $\Sigma$  e um alfabeto de saída Z.
- Estado interno do sistema descrito em termos de um conjunto de estados possíveis:
  - -Q
- Função de transformação do sistema descrita com base em duas funções distintas δ e λ:
  - Função de transição de estado:
    - $\delta: Q \times \Sigma \to Q$
  - Função de saída:
    - $\lambda: Q \times \Sigma \to Z$



Um modelo formal de computação pode ser descrito como um quíntuplo  $(Q, \Sigma, Z, \delta, \lambda)$ , onde:

- -Q é o **conjunto de estados** que caracterizam o sistema.
- $-\Sigma$  é o conjunto de símbolos de entrada (o **alfabeto de entrada**).
- -Z é o conjunto de símbolos de saída (o *alfabeto de saída*).
- $-\delta: Q \times \Sigma \to Q$  é a função de transição de estado.
- $-\lambda: Q \times \Sigma \to Z$  é a função de saída.

# Este tipo de modelo descreve um mecanismo computacional designado *Máquina de Estados*:

- A sua implementação física implica que o número de estados possíveis seja finito.
- Máquinas de Estados Finitos.

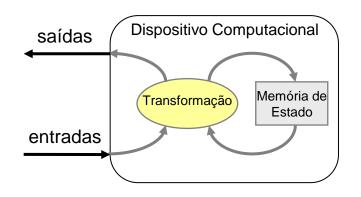
### Duas formulações distintas da função de saída *λ*:

 Máquinas de Mealy, nas quais a função de saída depende das entradas, ou seja:

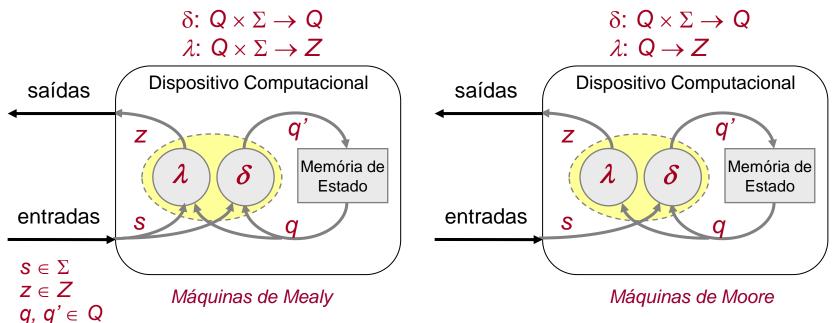
$$\lambda: Q \times \Sigma \to Z$$

 Máquinas de Moore, nas quais a função de saída não depende das entradas, ou seja:

$$\lambda: O \to Z$$



- Modelos sem estado
  - Presente
- Modelos com estado
  - Passado, presente



## Exemplo

### Sistema de Regulação Automática de Temperatura

O sistema de controlo recebe do exterior uma entrada que pode assumir valores com as seguintes representações simbólicas:

• T\_REG : indica que a temperatura está dentro dos limites definidos;

• T BAIXA : indica que a temperatura está abaixo do limite mínimo;

• T\_ALTA : indica que a temperatura está acima do limite máximo.

Por sua vez, o sistema produz uma saída para controlo dos mecanismos de aquecimento e de arrefecimento, a qual pode assumir valores com as seguintes representações simbólicas:

AQ : sistema de aquecimento é activado;

AR : sistema de arrefecimento é activado.

Na ausência dos valores AQ ou AR à saída do sistema de controlo, os mecanismos de aquecimento e de arrefecimento respectivos mantêm-se inactivos.

O sistema de controlo é caracterizado por três estados:

•  $q_{\text{inactivo}}$  : os sistemas de aquecimento e de arrefecimento estão inactivos;

•  $q_{\text{aquecimento}}$ : apenas o sistema de aquecimento está activo;

•  $q_{\text{arrefecimento}}$ : apenas o sistema de arrefecimento está activo.

## Exemplo

### Da descrição do problema podemos identificar:

• Um conjunto de símbolos de entrada (o alfabeto de entrada):

$$\Sigma = \{ T_REG, T_BAIXA, T_ALTA \}$$

• Um conjunto de símbolos de saída (o alfabeto de saída):

Um conjunto de estados que caracterizam o sistema de controlo:

$$Arr Q = \{ q_{\text{inactivo}}, q_{\text{aquecimento}}, q_{\text{arrefecimento}} \}$$

### Função de transição de estado:

$$\delta: \mathbf{Q} \times \Sigma \to \mathbf{Q}$$

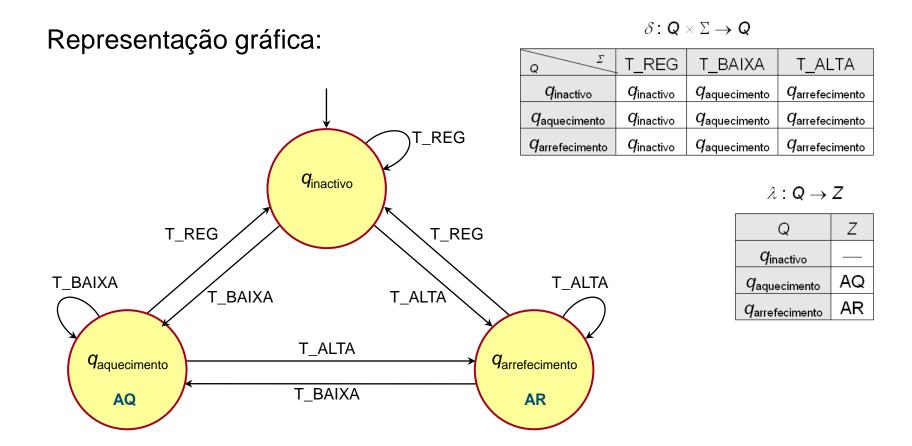
$Q$ $\Sigma$	T_REG	T_BAIXA	T_ALTA
<i>q</i> <sub>inactivo</sub>	$q_{inactivo}$	$q_{ m aquecimento}$	$q_{ m arrefecimento}$
<i>Q</i> aquecimento	$q_{inactivo}$	$q_{ m aquecimento}$	<i>q</i> arrefecimento
<i>q</i> <sub>arrefecimento</sub>	$q_{inactivo}$	$q_{ m aquecimento}$	<i>q</i> arrefecimento

### Função de saída:

$$\lambda: \mathbf{Q} \to \mathbf{Z}$$

Q	Z
<i>q</i> <sub>inacti∨o</sub>	
$q_{ m aquecimento}$	AQ
<i>Q</i> arrefecimento	AR

# Diagrama de Transição de Estado

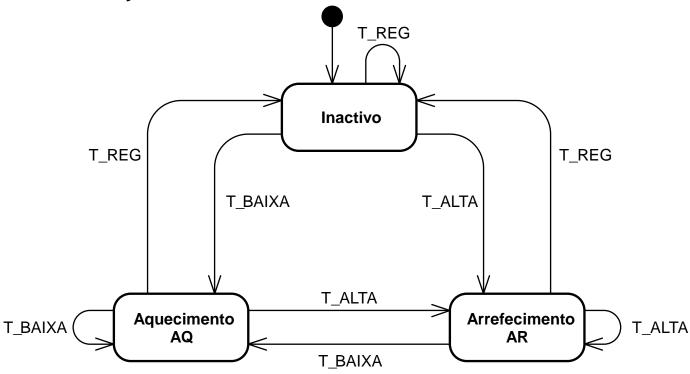


Latência de transição?

# Diagrama de Transição de Estado

### Linguagem UML

Diagramas de Transição de Estado



ISEL-DEETC, ES0708, L. Morgado, 2008

# Diagrama de Transição de Estado

### Estado

Representação de situação de evolução de um sistema ou parte de um sistema

### Transição

Acontecimento através do qual o sistema evolui do estado actual para um novo estado

#### Evento

Ocorrência no tempo e no espaço com significado para a evolução de estado

### Guarda

- Condição que inibe ou permite transições ou acções
  - Exemplo típico de utilização de variáveis de estado

### Acção

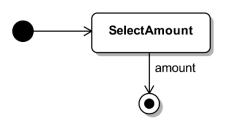
- Define comportamento
  - Associado a transição
  - Associado a estado

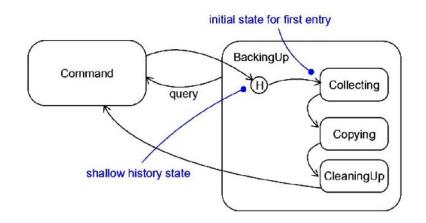


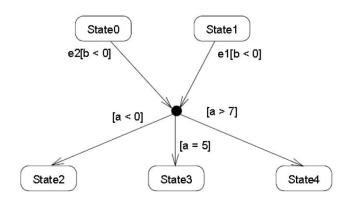
## Pseudo-estados

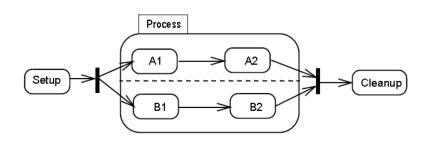
- Símbolos utilizados com significado específico para definição de semântica adicional - vértices transientes do grafo de transição de estado
  - Início: Representa a fonte de transição inicial da máquina
  - Fim: Representa o destino para a transição final da máquina
  - Histórico: Representa o sub-estado (superficial shallow) ou estado (profundo deep) mais recentemente activo de um estado composto
    - **Bifurcação / Reunião**: Representam a separação/reunião de transições envolvendo diferentes regiões ortogonais
  - Junção: Suporte geral para relacionar diferentes transições sem introduzir semântica adicional

## Pseudo-estados



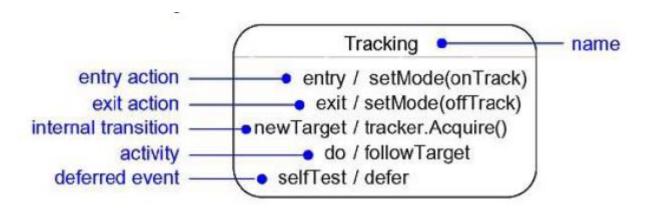






[OMG, 2020; UML User Guide, Booch et al. 1998]

## Acções de Estado



1. Name	A textual string that distinguishes the state from other states; a state may be anonymous, meaning that it has no name
2. Entry/exit actions	Actions executed on entering and exiting the state, respectively
3. Internal transitions	Transitions that are handled without causing a change in state
4. Substates	The nested structure of a state, involving disjoint (sequentially active) or concurrent (concurrently active) substates
5. Deferred events	A list of events that are not handled in that state but, rather, are postponed and queued for handling by the object in another state

[UML User Guide, Booch et al. 1998]

## Acções de Entrada e de Saída

## entry

Acções executadas após a entrada no estado

### exit

- Acções executadas antes da saída do estado
- Iniciação e finalização de estado

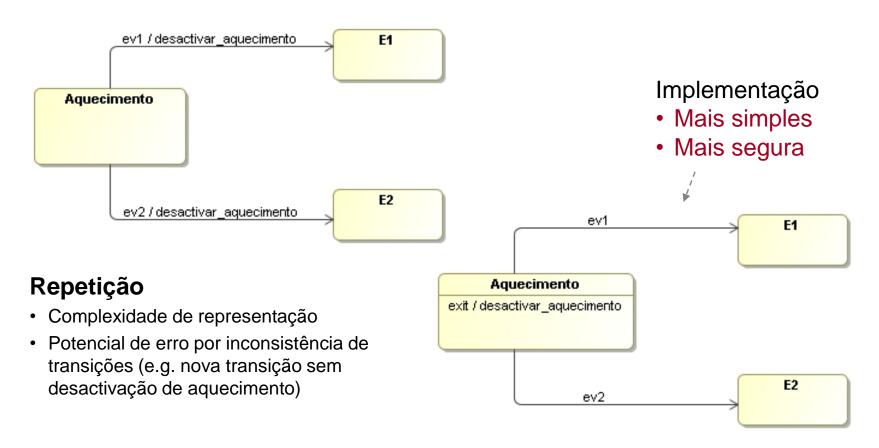
TypingPassword

entry/setEchoInvisible() exit/setEchoNormal() character/handleCharacter() help/displayHelp()

## Acções de Entrada e de Saída

### Exemplo: Forno de cozinha

 Requisito crítico de segurança: desactivar sistema de aquecimento sempre que exista um acesso ao forno



# Bibliografia

[Booch et al., 1998]

G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, UML 2 Toolkit, Wiley, 2004.

[OMG, 2020]

Unified Modeling Language (Specification), OMG, 2020.