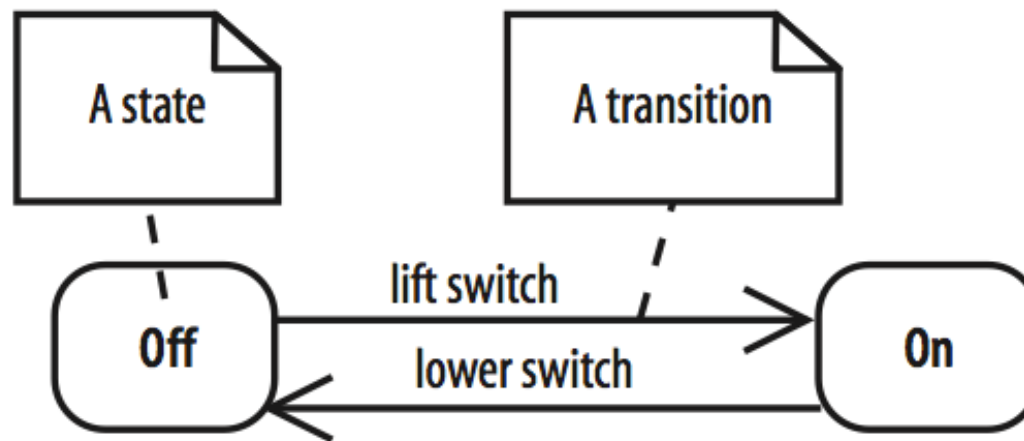
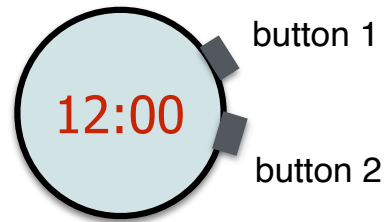


# Diagramas de Estados (UML)

- Lo esencial : estados y transiciones entre estados

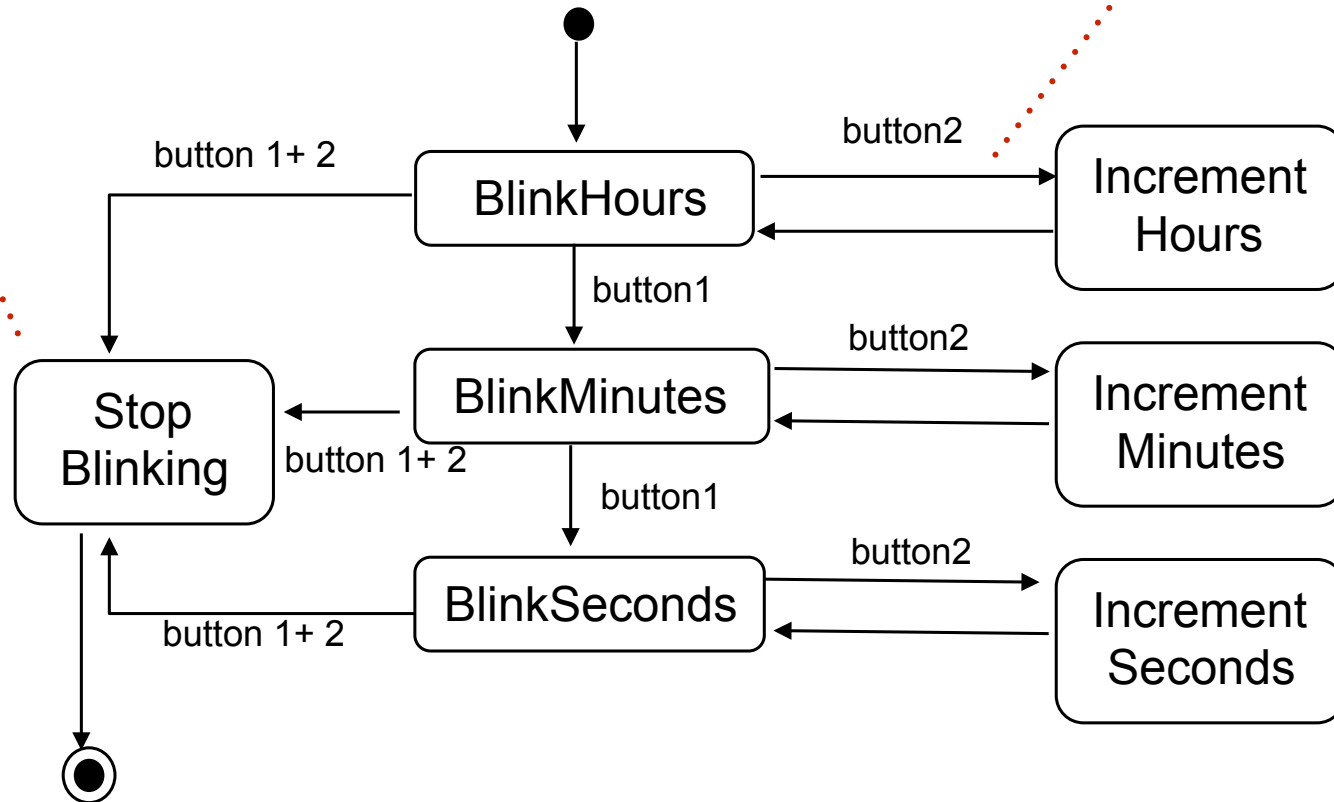


# Ejemplo

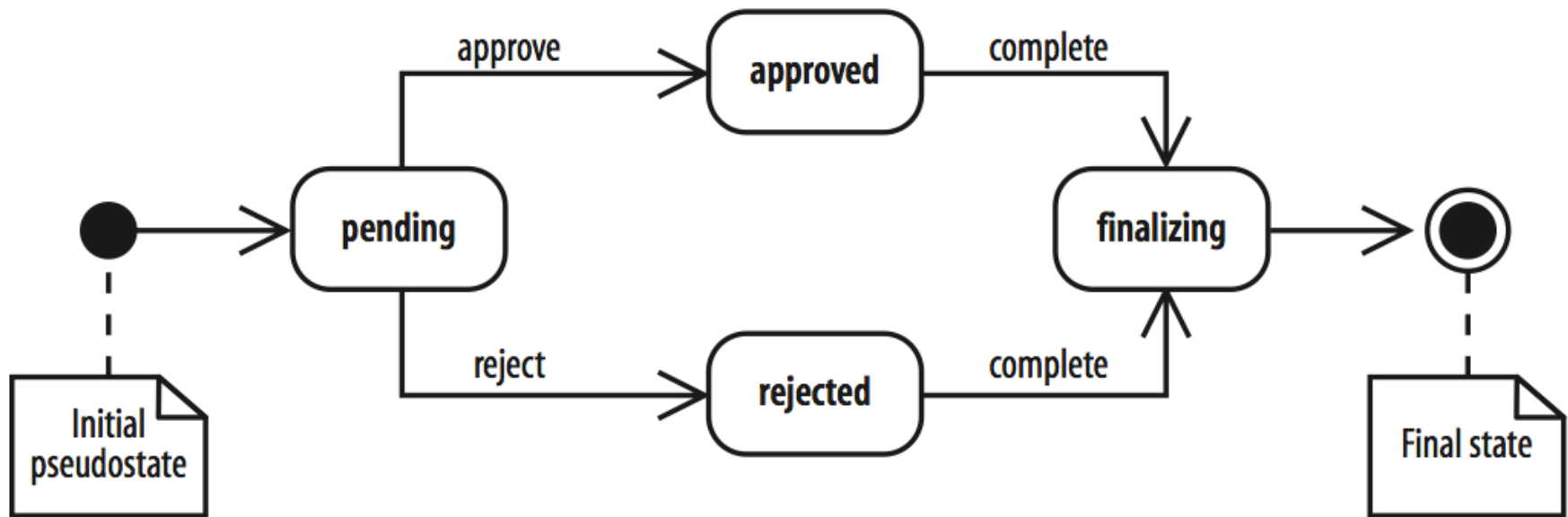


**estado**

**transición**

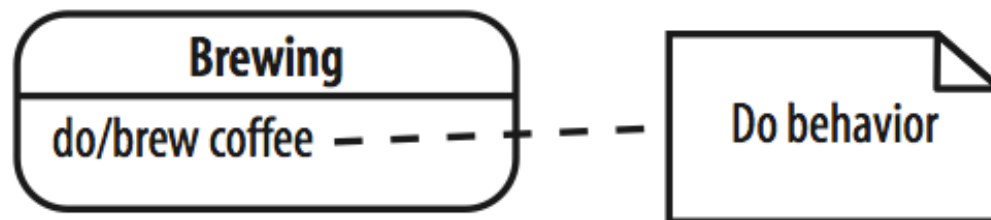


# Estados Inicial y Final

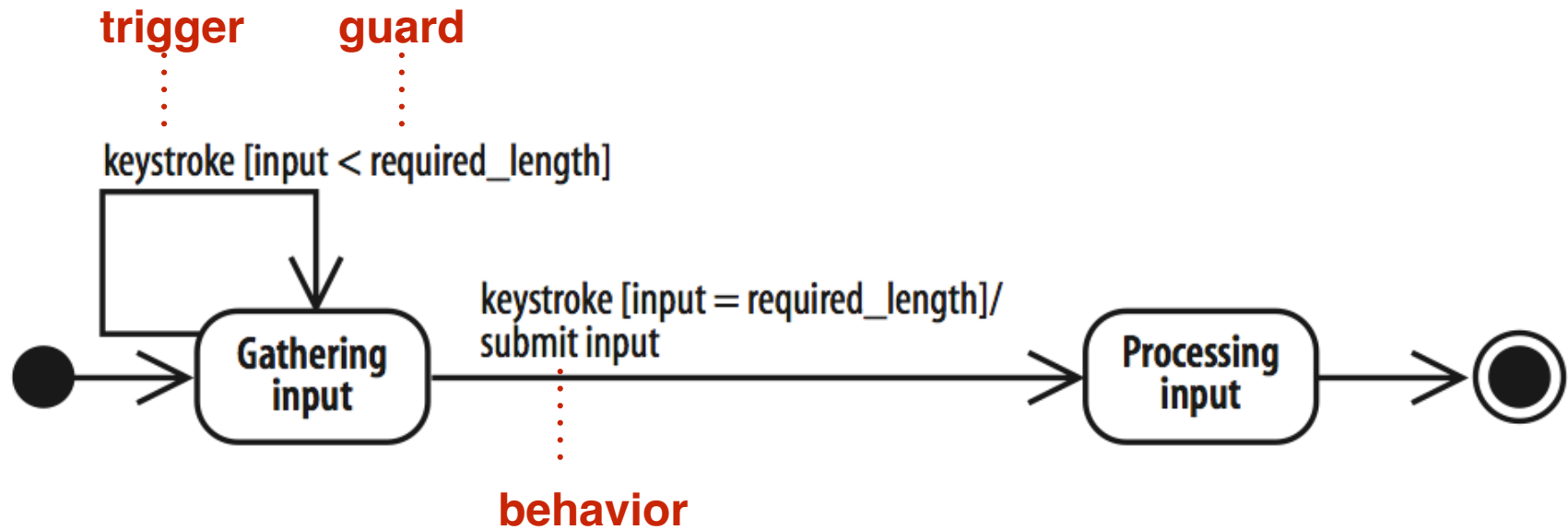


# Estados Activos

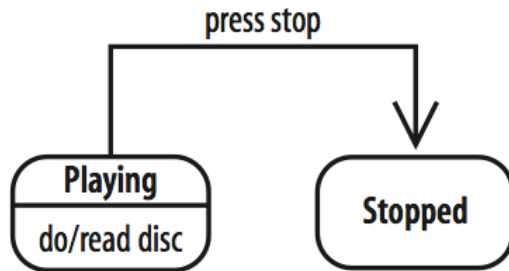
- ▶ Un estado pasivo es por ejemplo on/off
- ▶ Un estado activo es algo que está ocurriendo o haciéndose (por ejemplo “calculando”)
- ▶ En este caso se puede describir comportamiento en el interior del estado (parte tan pronto se ingresa al estado)



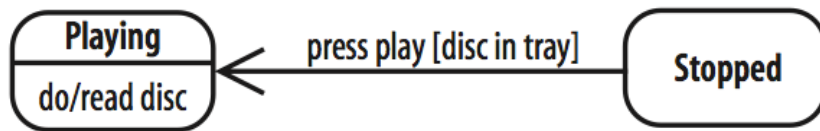
# Transiciones



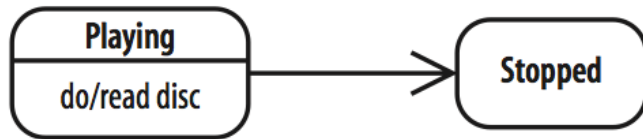
# Variaciones



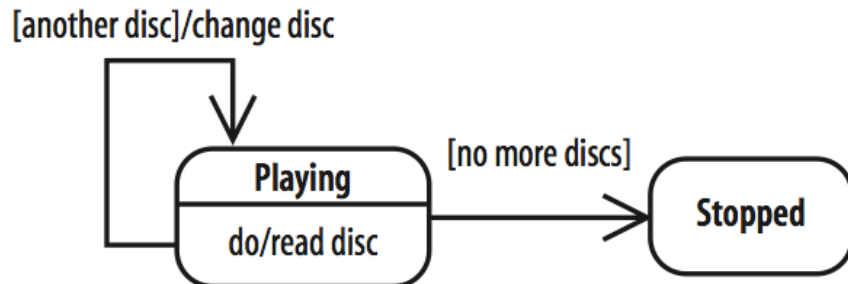
**trigger simple**



**trigger con guard**



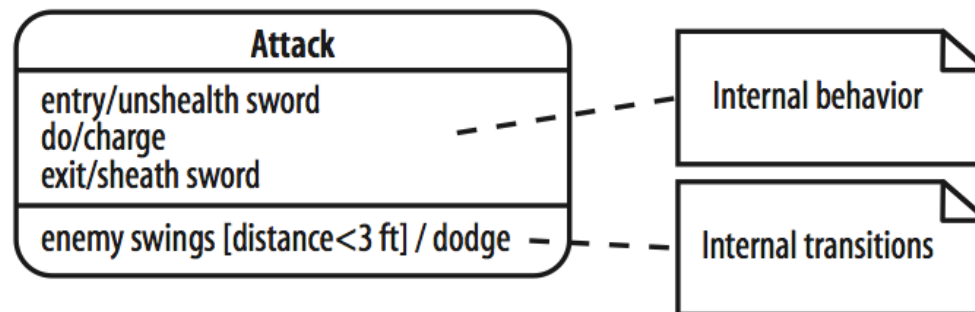
**transición automática**



**guards para elección**

# Mas sobre comportamiento durante un estado

- ▶ aparte de “do” puede especificarse un “entry” y un “exit”
- ▶ entry behaviour - sucede tan pronto se activa estado
- ▶ exit behaviour - sucede inmediatamente antes de salir de ese estado
- ▶ transición interna - reacción dentro del estado



# Componentes, Módulos, Clases

- ▶ Un diseño debe poder entenderse
- ▶ Complejidad hacer necesario descomponer la solución en unidades menores
- ▶ Modularidad es clave en lograr ese objetivo
- ▶ A menudo se requiere descomposición jerárquica: 5 subsistemas, cada uno con 4 componentes, cada una con una cierta estructura de clases
- ▶ Muchas posibles descomposiciones, ¿cual elegir?