

# *UML State Machine Diagrams* Máquinas Secuenciales

Alejandro Sánchez

Departamento de Informática  
Universidad Nacional de San Luis

Universidad Tecnológica Nacional  
Facultad Regional Tucumán  
14-15 Junio

# Contenidos

## *UML state machine diagrams* Máquinas secuenciales

- Esencia
- Constructores
- Relaciones con otros diagramas

# Taxonomía de diagramas UML

## Diagrams

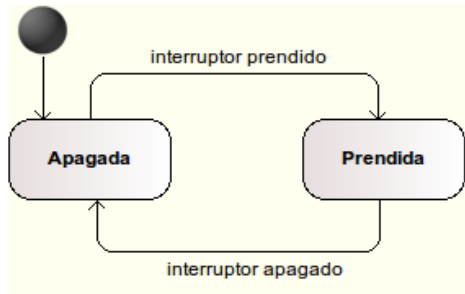
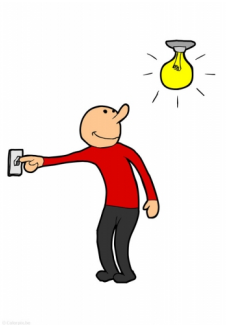
- Behaviour
  - Use cases
  - Activity
  - State machines
- Interaction
  - Sequence
  - Communication
  - Interaction overview
  - Timing
- Structure
  - Class
  - Composite structure
  - Components
  - Deployment
  - Objects
  - Packages

# ¿Qué es una *state machine* ?

## State machine (Máquina de estados)

Describe como un objeto o sistema  
cambia en el tiempo,  
a que eventos responde,  
y como lo hace

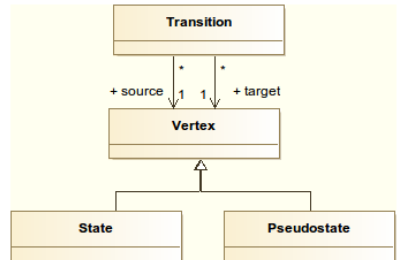
# Ejemplo: luz e interruptor



# Elementos principales del metamodelo – *State*

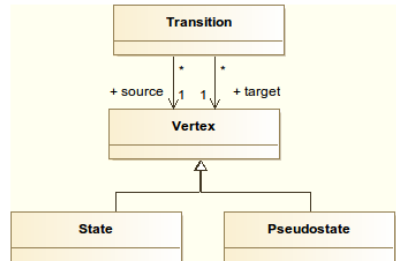
- **State** (Estado)  
modela una situación  
durante la cual algún  
(posiblemente implícito)  
invariante se cumple.

Ej.: *Apagada*



# Elementos principales del metamodelo – *Transition*

- **Transition** (Transición)  
relación dirigida entre un estado origen y un estado destino, representando como se responde a la ocurrencia de un evento cuando se encuentra en el estado de origen



Ej.: *Interruptor prendido*

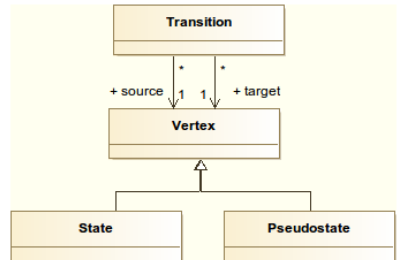
# Elementos principales del metamodelo – *Pseudostate*

- **Pseudostate**

(Pseudoestado)

Marcadores que dirigen el flujo en una máquina de estados

Ej.: *Marcador de estado inicial*





# ¿A qué llamamos máquina secuencial?

Una posible caracterización ...

Máquina secuencial

Máquina de estado  
que en cualquier instante en el que se la observe  
tiene un único estado activo

Ejemplos

- Una luz con interruptor (Secuencial)
- Un elevador y su sistema de puertas (No secuencial)

# Tipos de máquinas de estado y sus aplicaciones

## Behavioural state machines

### Modelar

- un sistema
- el ambiente de un sistema
- un objeto
- una operación

## Protocol state machines

Especificar el patrón de  
mensajes esperados  
por un objeto

# *Behavioural state machines*

- Modelado de un sistema  
Una máquina expendedora de bebidas
- Modelado del ambiente de un sistema  
Investigador del departamento de informática
- Modelado de un objeto  
El carro de compras de un sitio web
- Modelado de una operación  
Actualizar la información de un cliente en distintos departamentos de una empresa

# *Protocol state machines*

El protocolo de comunicación del objeto carrito de compras  
en un sistema de venta on-line

El protocolo de comunicación del objeto que coordina  
la actualización de la información de un cliente  
con cada uno de los sistemas  
de los departamentos de una empresa

# Constructores de máquinas de estado

- Transition
  - Event
  - Guard
  - Action
  - Postcondition
- State
  - Entry behaviour
  - Do behaviour
  - Exit behaviour
  - Internal transition
  - Final state
- Pseudo state
  - Initial
  - Choice
  - ...

# Sintaxis de la etiqueta de transición

Behavioural state machine

*event [guard] / action*

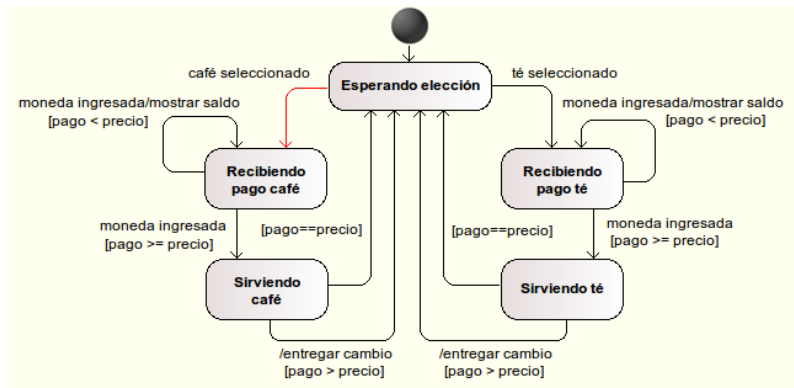
Protocol state machine

*event [guard] {postcondition}*

Todos son opcionales

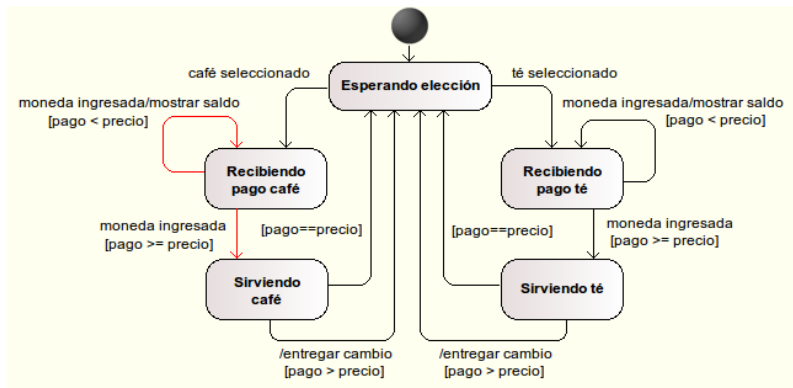
# Event

Es un disparador explícito de la transición



# Guard

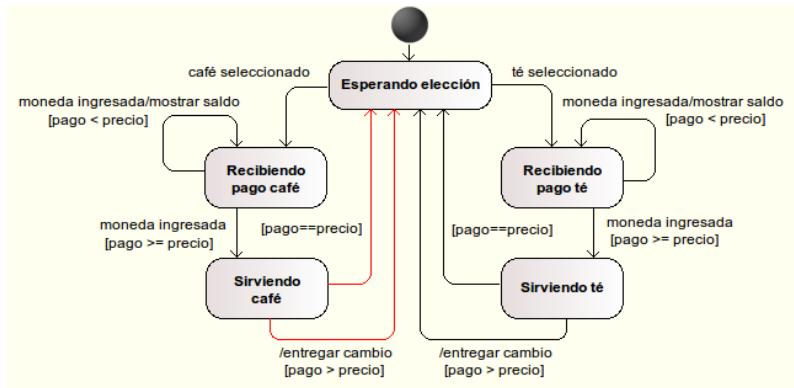
Expresión booleana que permite o bloquea la transición





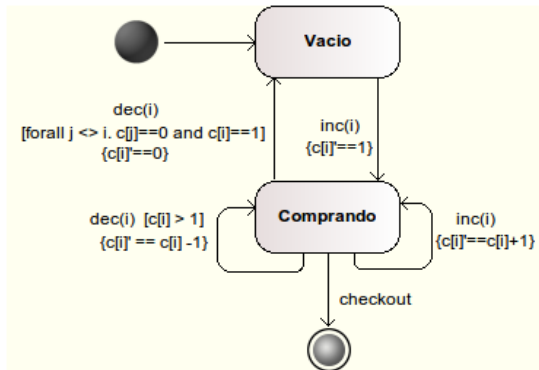
# Action

Actividad no interrumpible  
que se ejecuta mientras la transición tiene lugar



# Postcondition

Se cumple una vez terminada la transición



# Sobre la semántica de las transiciones

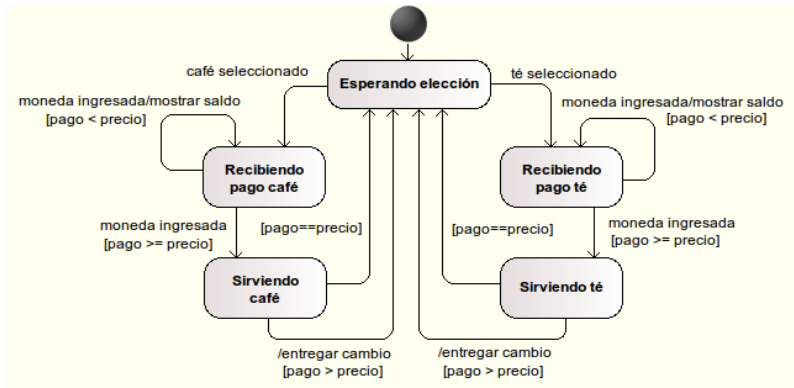
¿Cómo se interpreta  
cuando no hay transición para un evento  
en un estado?

Se puede interpretar de tres formas

- Skip. No tiene efecto en el estado
- Error. Tiene lugar y causa un efecto inesperado
- Blocking. No puede acontecer

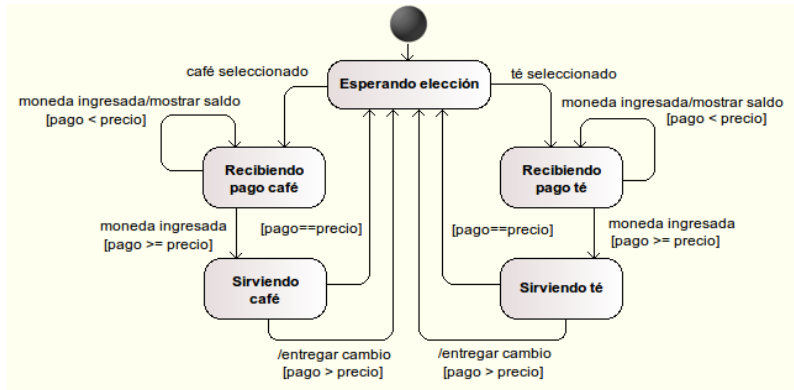
# Sobre la semántica de las transiciones - Skip

Estado *Sirviendo té*  
recibe evento *Café seleccionado*



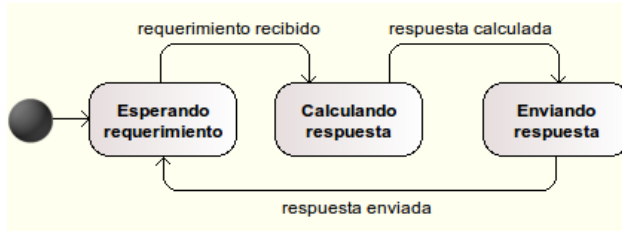
# Sobre la semántica de las transiciones - Error

Estado *Sirviendo té*  
recibe evento *Moneda ingresada*



# Sobre la semántica de las transiciones - Blocking

Un cliente realiza un requerimiento  
a un servidor en estado *Calculando respuesta*

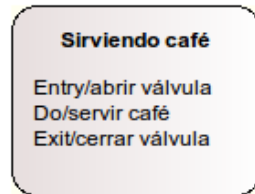


# Comportamiento en estados

Entry behaviour / Do behaviour / Exit behaviour

Para el estado sirviendo café

- Entry behaviour:  
abrir válvula
- Do behaviour:  
servir café
- Exit behaviour:  
cerrar válvula

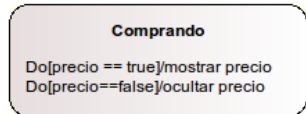


# Transición interna a estados

event [guard ] action

Para el estado comprando del carrito de compras

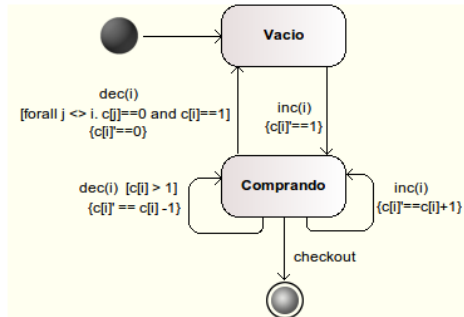
- precio seleccionado /  
calcular y mostrar precio
- precio deseleccionado /  
ocultar precio





# Estado final

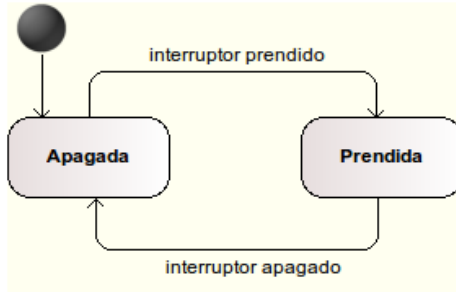
## Carrito de compras



El carrito no se utiliza para otro cliente.

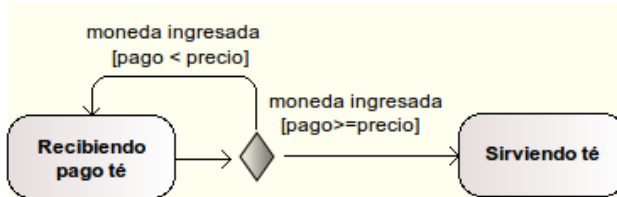
# Initial state

Marca el inicio de la máquina de estados



# Choice

Hacer evidente una bifurcación de acuerdo a una condición

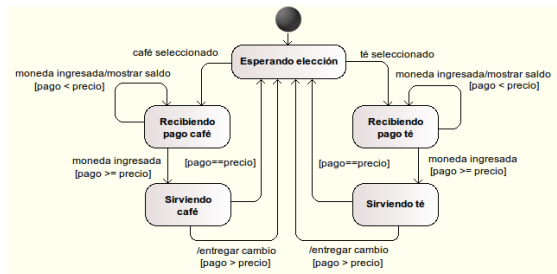


# Relación con una clase

Puede utilizarse una *protocol state machine*  
y luego refinarse a una *behavioural state machine*

Un método puede ser un *event* o una *action*

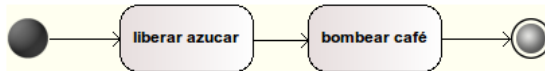
Un atributo puede aparecer en un *guard* o una *postcondition*



Máquina Expendedora
pago : string
moneda ingresada() mostrar saldo() cafe seleccionado() te seleccionado() entregar cambio() servir café() servir té()

# Relación con un método

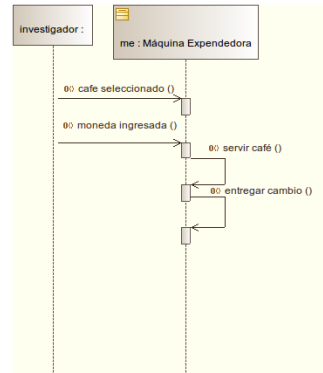
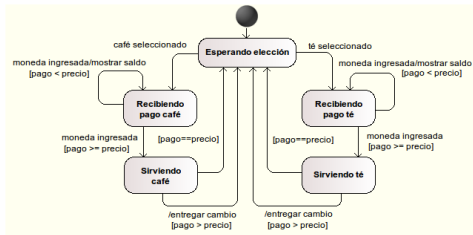
método servir café



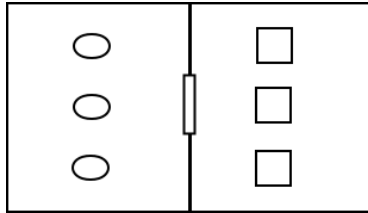
Máquina Expendedora
pago : string
moneda ingresada() mostrar saldo() cafe seleccionado() te seleccionado() entregar cambio() servir café() servir té()

# Relación con un diagrama de secuencia

Un escenario posible para la máquina expendedora de bebidas



# Problema de luces e interruptores

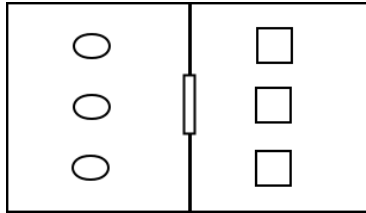


Dos habitaciones y una puerta maciza que las comunica.  
Tres luces en una habitación y tres interruptores en la otra.  
Los interruptores solo pueden usarse hasta abrir la puerta.

¿Qué luz corresponde a que interruptor?

Tip: el modelo luz-interruptor ya visto no sirve

# Problema de luces e interruptores



Dos habitaciones y una puerta maciza que las comunica.  
Tres luces en una habitación y tres interruptores en la otra.  
Los interruptores solo pueden usarse hasta abrir la puerta.

¿Qué luz corresponde a que interruptor?  
Tip: el modelo luz-interruptor ya visto no sirve



# Problema de integración de sistemas I

## Capacitación profesional

La empresa cuenta con tres sistemas independientes: sistema de cursos, un CRM, y un ERP. La información de un cliente se actualiza a través de scripts.

### Mejora

Se define un servicio para permitir el acceso a cada subsistema. Se define un servicio coordinador que integra la actualización del cliente en cada sistema.

### Tarea:

Realizar un diagrama de máquinas de estado para el coordinador. Considere que el cliente puede no existir y que un servicio puede fallar.

# Problema de integración de sistemas II

¿Qué problema ve en esta máquina de estados?

