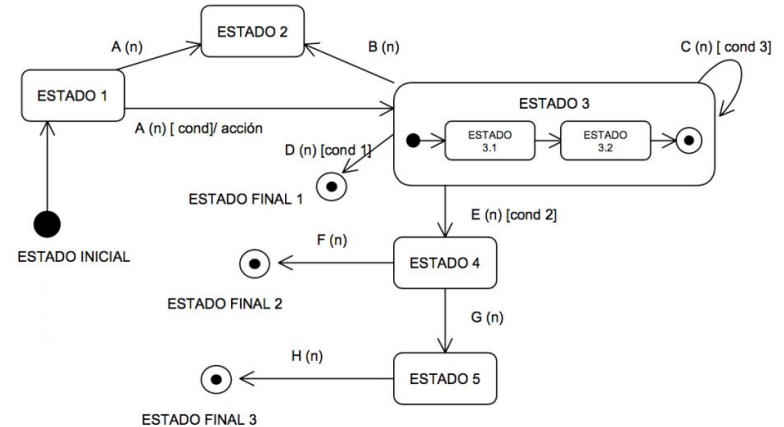


# UML Diagramas de estado



# Índice

[Introducción](#)

[Transiciones](#)

[Transición simple](#)

[Transición de finalización \(sin evento\)](#)

[Transición interna](#)

[Transición compleja](#)

[Estados](#)

[Acciones de los estados](#) (entry, exit, do)

# ¿Para qué sirven?

Son útiles para modelar la vida de un objeto

Un diagrama de estados muestra el flujo de control entre estados (en qué estados posibles puede estar “algo” y como se producen los cambios entre dichos estados)

En el diagrama de estados se indica que eventos hace que se pase de un estado a otro, y cuales son las respuestas y acciones que genera. También ilustra que eventos pueden cambiar el estado de los objetos de la clase.

# No confundir entre diagrama de estado y diagrama de actividad

Los **Diagramas de Actividad** son muy parecidos a los DF (diagramas de Flujo) pero la diferencia radica en que todas las actividades están unidas de forma clara a objetos.

Los **Diagramas de Estado** muestran los diferentes "estados" de un objeto durante su ciclo de vida como también las acciones que generan estos cambios en el objeto.

Por tanto un diagrama de actividad es una variación el diagrama de estado, donde cada "estado" representa a las distintas operaciones y cada transiciones representa las diferentes actividades que ocurren cuando la operación es completa.

# Máquina de Estado

Los Diagramas de Estados muestran una Máquina de Estado.

Una máquina de estados es un comportamiento que especifica las secuencias de estados por las que pasa un objeto a lo largo de su vida en respuesta a eventos, junto con sus respuestas a esos eventos

*(Booch, Rumbaugh, Jacobson)*

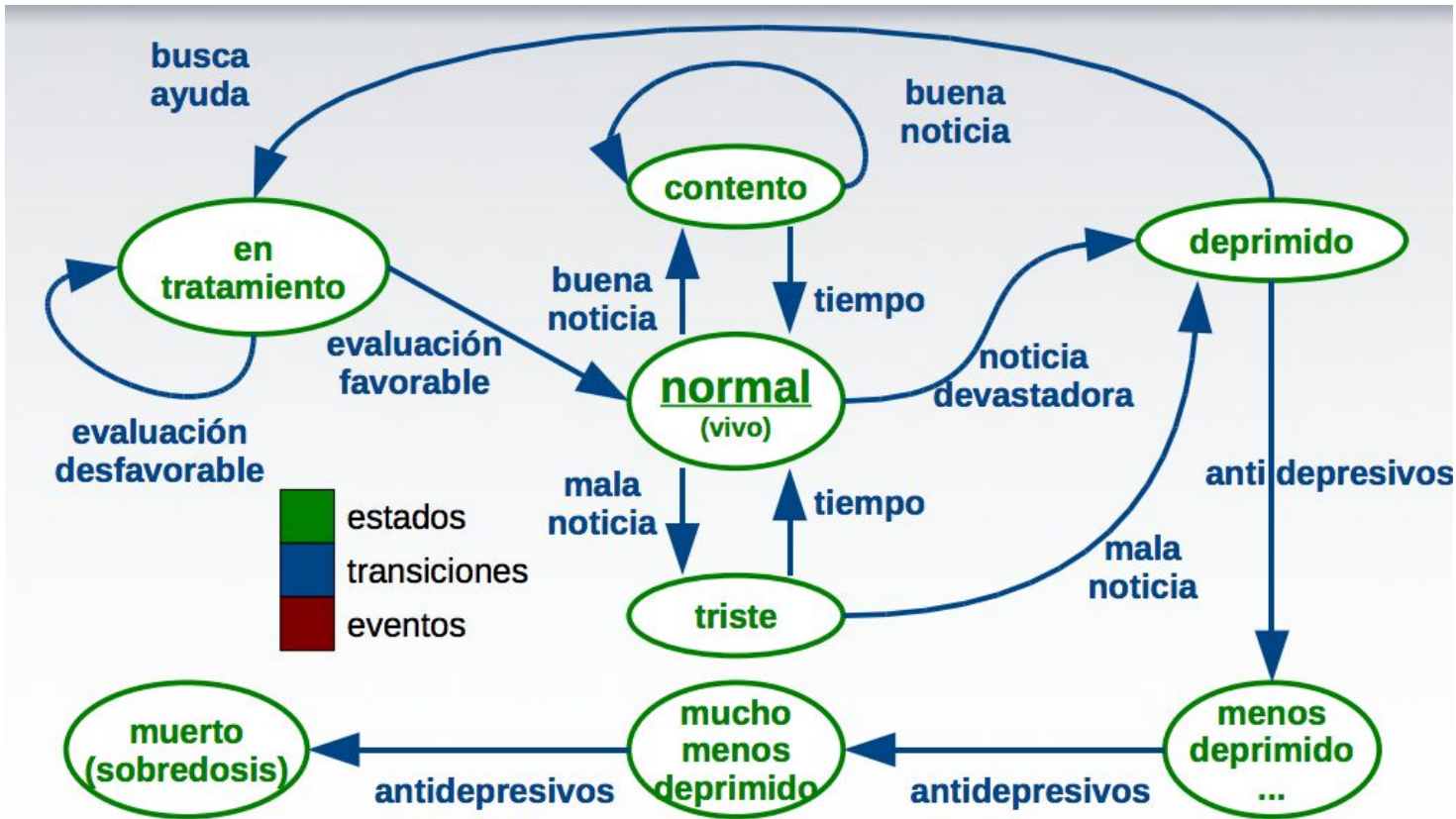
# Estados / Eventos / Transiciones

Un **estado** es una **condición o situación** en la vida de un objeto durante la cual **satisface una condición, realiza alguna actividad o espera algún evento**.

Un **evento** es la especificación de un **acontecimiento significativo** que ocupa un lugar en el **tiempo** y en el **espacio**. Es la aparición de un estímulo que puede (o no) activar una transición de estado

Una **transición** es una **relación** entre dos estados que indica que un objeto que esté en el **primer estado** realizará ciertas acciones y entrará en el **segundo estado** cuando ocurra un evento especificado y se satisfagan unas **condiciones** especificadas

¿En qué estado (de ánimo) se encuentra usted y cómo cambia su estado de ánimo?



Nota: Esto no es un diagrama de estados real de UML, es sólo un ejemplo

Ángel González M.

# Estado

Identifica un **periodo de tiempo del objeto** (no instantáneo) en el cual el objeto está esperando alguna **operación**, tiene cierto estado característico o puede recibir cierto tipo de estímulos.

Estas ocurrencias pueden ser por:

- Condición que obtiene el valor de verdadero o falso
- Recepción de una señal de otro objeto en el modelo
- Recepción de un mensaje
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular



Estado

*Ángel González M.*



# Ejemplo

Diagrama de clases de una lavadora

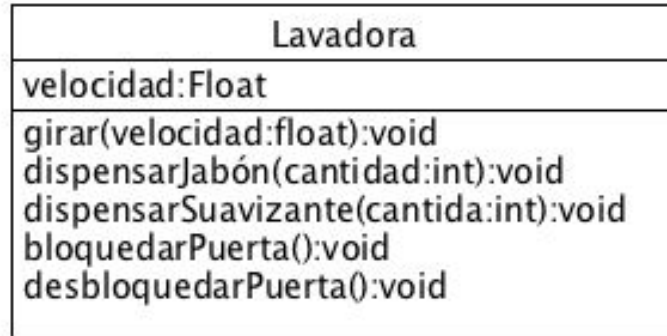
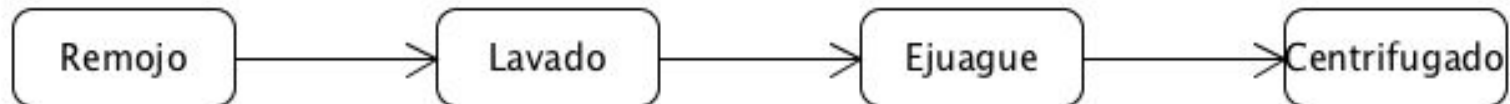


Diagrama de estados de una lavadora



# Punto de inicio y fin

inicio



fin



# Evento

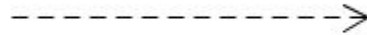
Es una ocurrencia que puede causar la transición de un estado a otro de un objeto



# Envío de mensajes

Además de mostrar la transición de estados por medio de eventos, se puede representar el momento en el cual se envían mensajes a otros objetos.

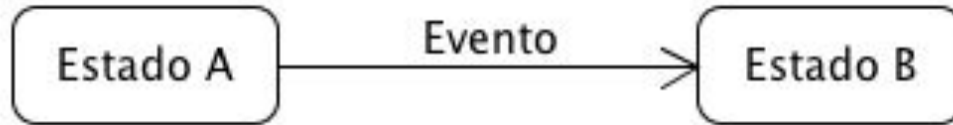
Se usa una línea discontinua



# Transiciones

# Transición

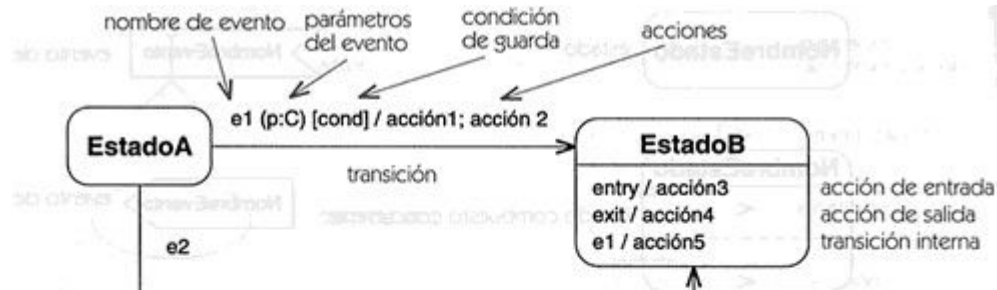
Se usa una línea continua



# Formato de las transiciones

Las transiciones se representan con el siguiente formato:

***nombre\_evento, [ condicion\_guarda ] / action , send-clause***



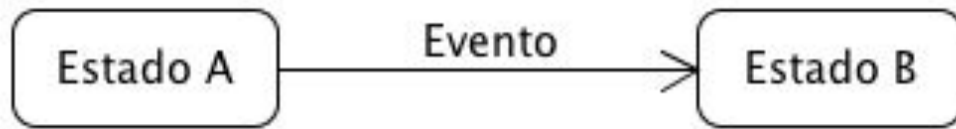
- **nombre\_evento** es la explicación del evento que da lugar a la transición.
- **condicion\_guarda** son las condiciones adicionales al evento, necesarias para que la transición ocurra.
- **Action** es un mensaje al objeto o a otro objeto que se ejecuta como resultado de la transición y el cambio de estado.
- **Send-clause** son operaciones adicionales que se realizan con el cambio de estado, por ejemplo, el envío de eventos a otros paquetes o clases.

Ángel González M.

# Transición

Tenemos 6 tipos:

- Transición simple
- Transición de finalización (sin evento)
- Transición interna
- Transición compleja
- Transición temporizada
- Transición a estados anidados



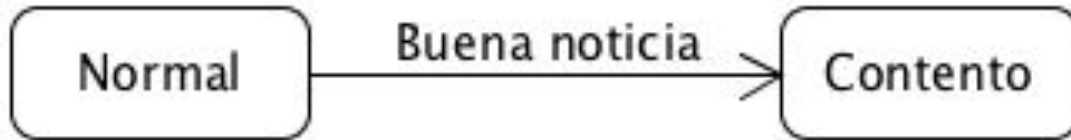
Se usa una línea continua



# Transición simple

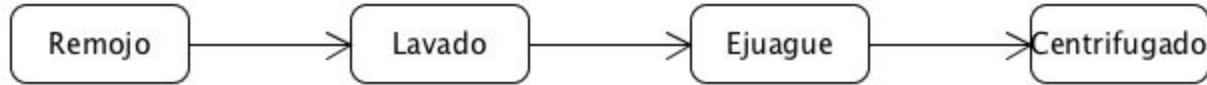
Es una relación entre 2 estados, que indica que un objeto en el primer estado (cuando ocurre un evento) puede entrar al segundo estado y ejecutar ciertas operaciones, y si ciertas condiciones son satisfechas.

Puede estar acompañada de un texto con el siguiente formato:



# Transición de finalización (sin evento)

Carece de un evento disparador, se dispara al finalizar la actividad



# Transición interna

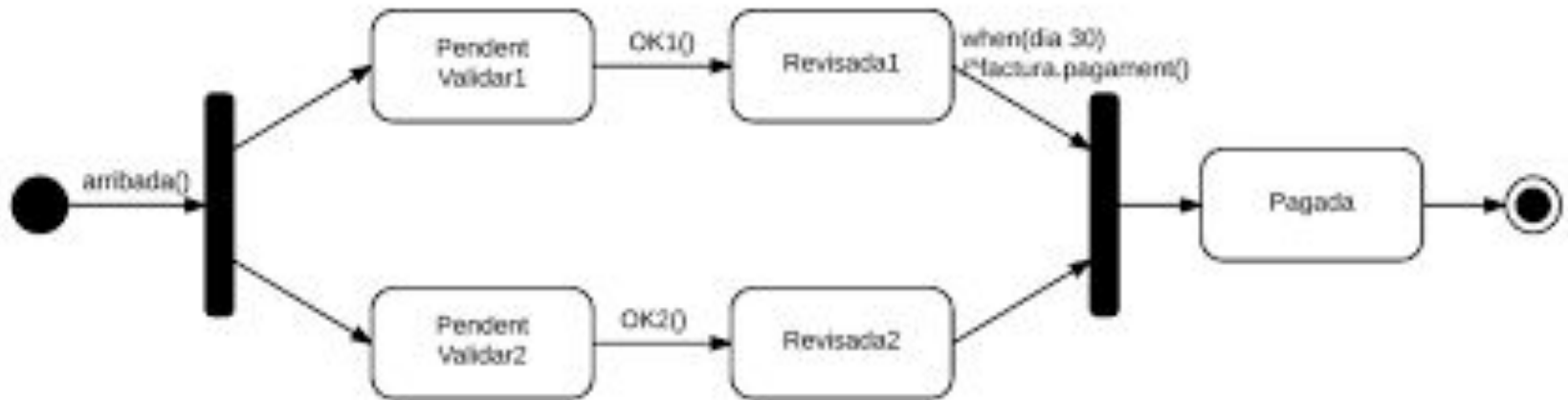
Es una transición que permanece en el mismo estado, en vez de involucrar dos estados distintos. Representa un evento que no causa cambio de estado. Se denota como una cadena adicional en el compartimiento de acciones del estado



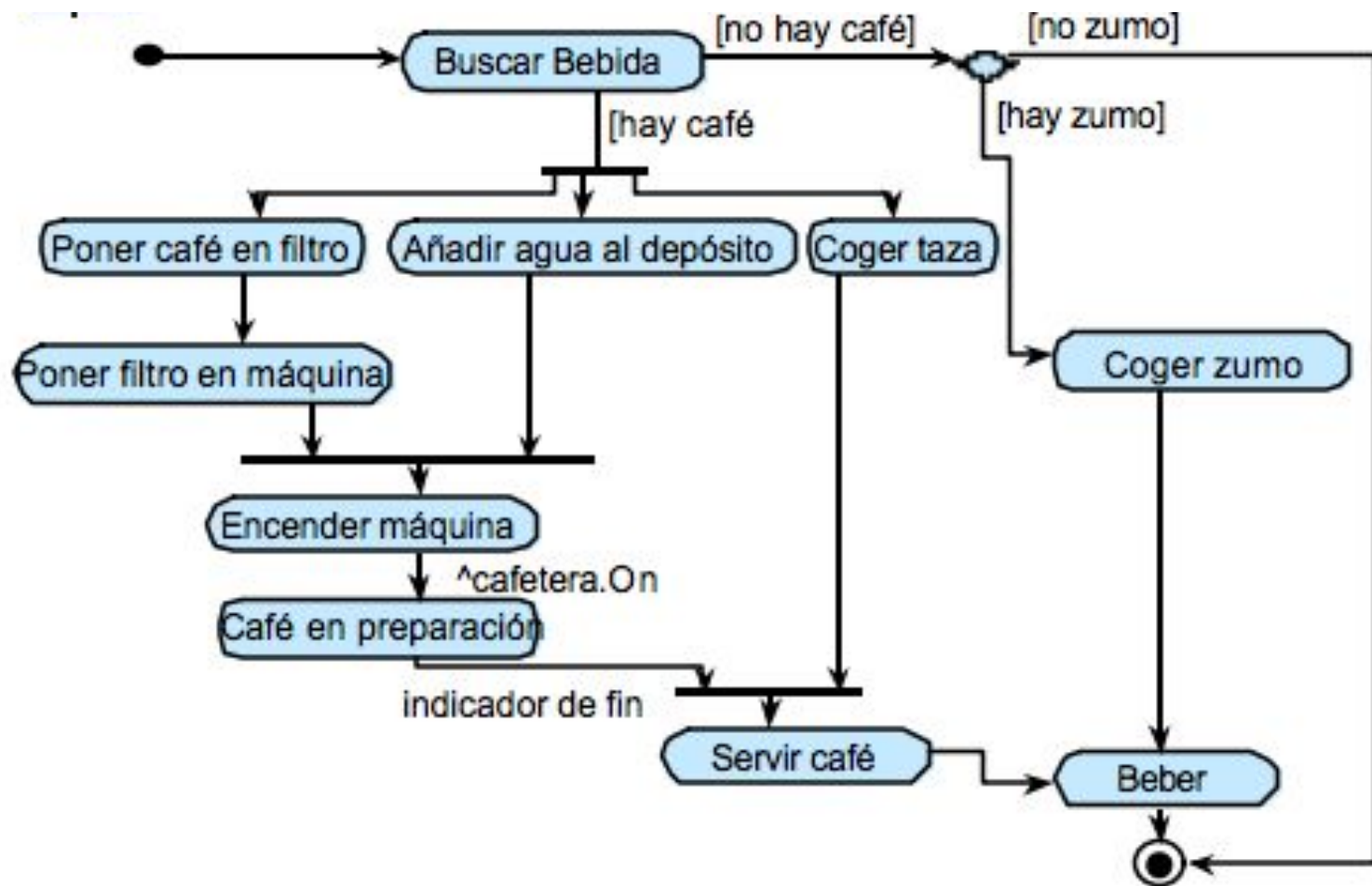
# Transición compleja

Relaciones dos o más estados en una transición de múltiples fuentes y/o múltiples destinos.

Representa la **subdivisión en threads** del control del objeto o una sincronización. Se representa como una línea vertical de la cual salen o entran varias líneas de transición de estado.



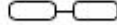
## Ejemplo



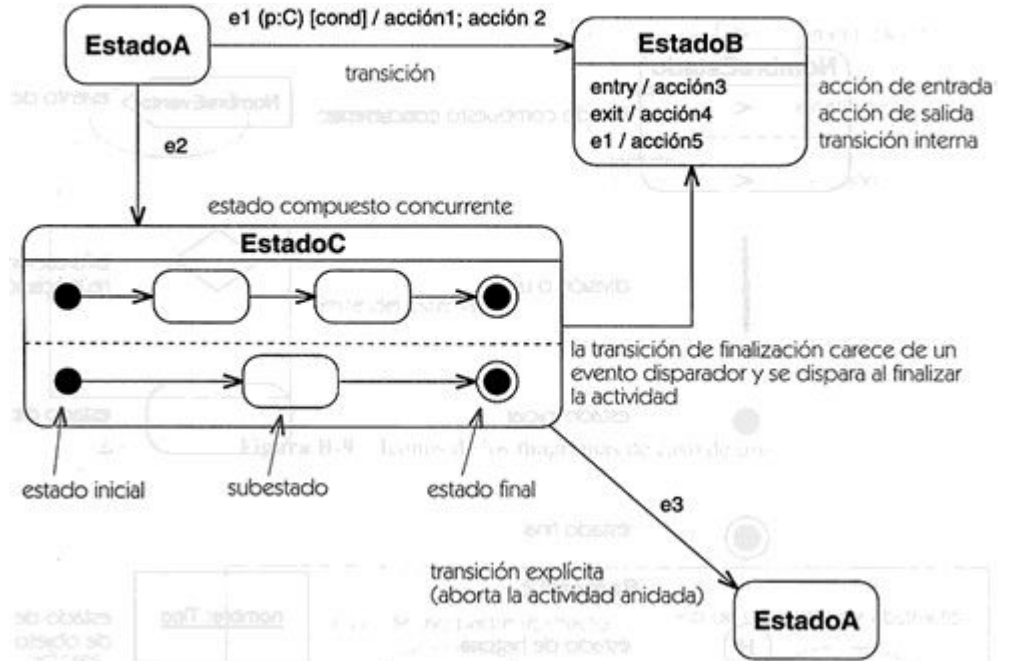
# Estados

# SubEstados

Estado con  
subestados



Un estado se puede descomponer en subestados, con transiciones entre ellos y conexiones al nivel superior. Las conexiones se ven al nivel inferior como estados de inicio y fin, los cuales se suponen conectados a las entradas y salidas del nivel inmediatamente superior.



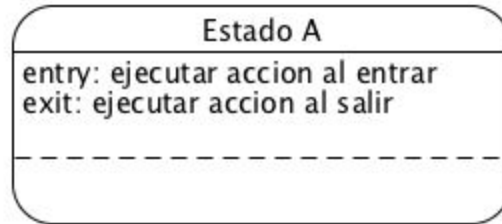
# Acciones (De los estados)

Podemos especificar la solicitud de un servicio a otro objeto como consecuencia de la transición. Se puede especificar el ejecutar una acción:

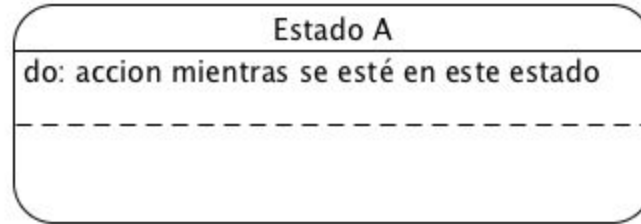
- como consecuencia de entrar en un estado (entry)
- como consecuencia de salir de un estado (exit)
- como consecuencia estar en un estado (do)
- o por la ocurrencia de un eventos



# Acciones (entry y exit)

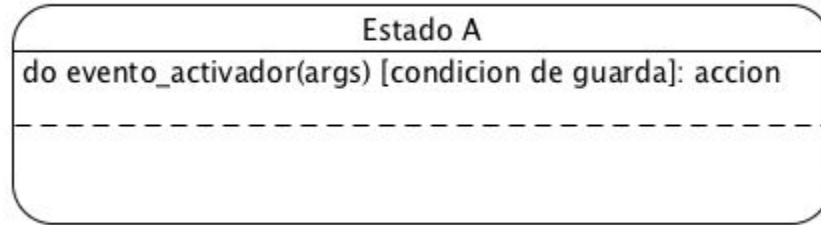


# Acciones (do)



# Acción (do on)

Se puede especificar el hacer una acción cuando ocurre en dicho estado un evento que no conlleva salir del estado

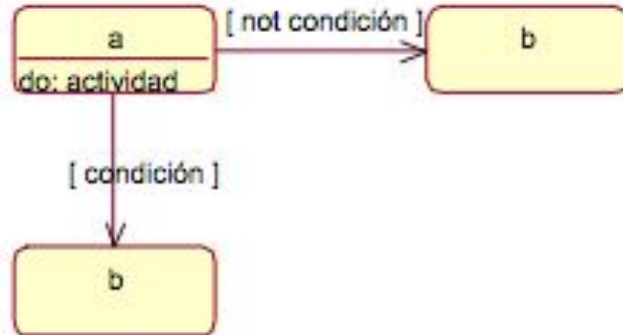


# Actividad

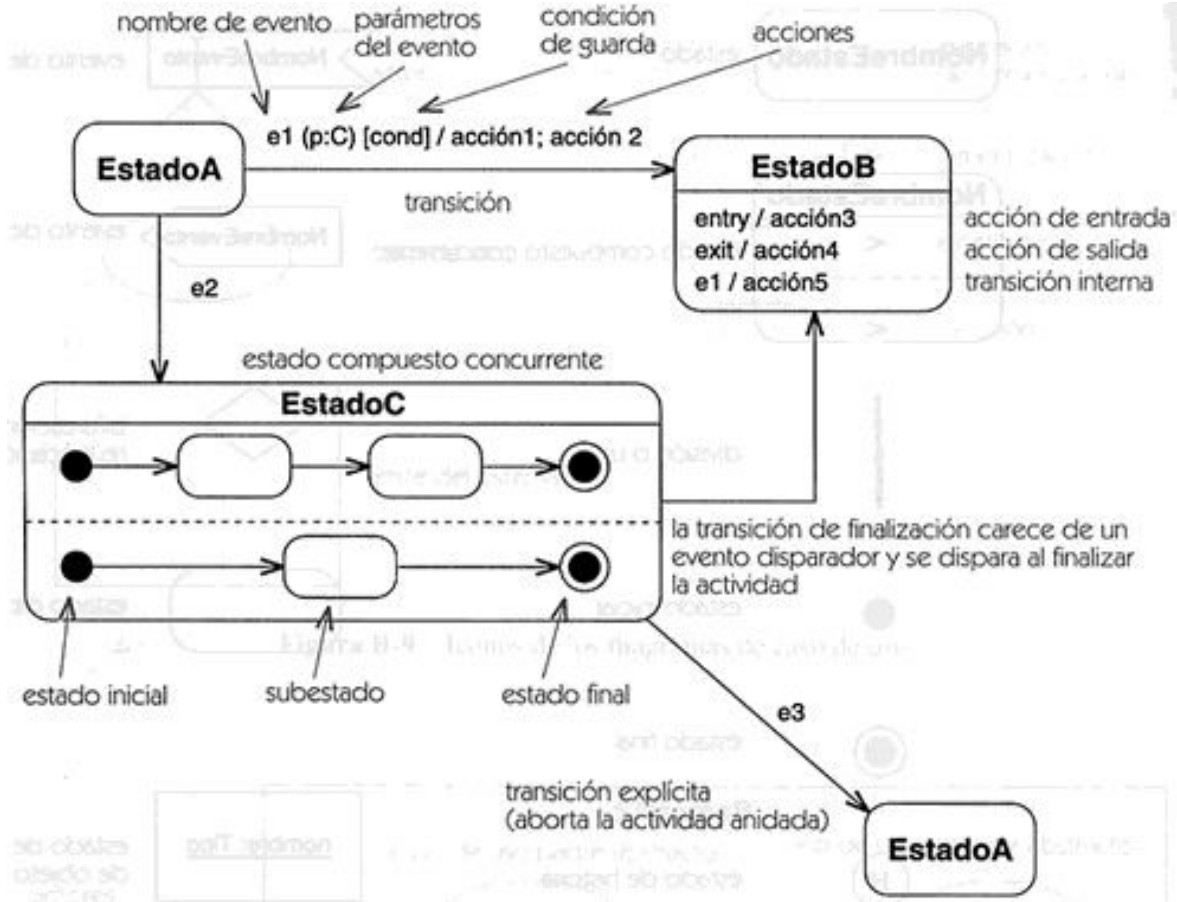
Las actividades son similares a las acciones pero tienen duración y se ejecutan dentro de un estado del objeto

Las actividades pueden interrumpirse en todo momento, cuando se desencadena la operación de salida del estado

Cuando una actividad finaliza se produce una transición automática de salida del estado



# Ejemplo completo



# Ejemplos sacados de internet

# Ejemplo

Diagrama de clases

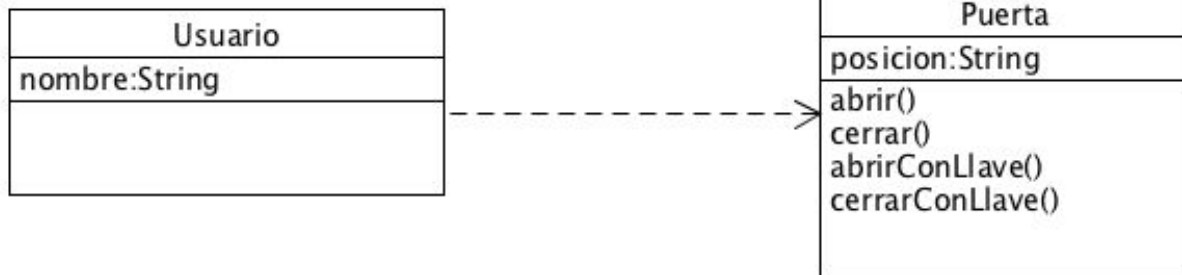
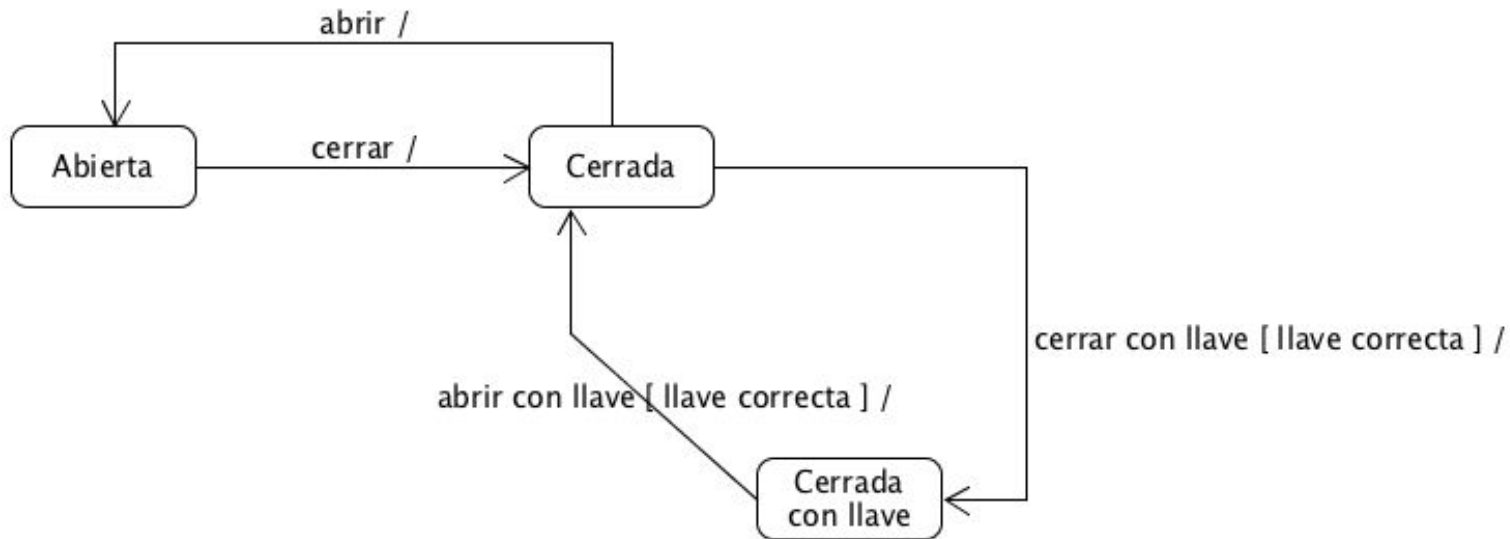


Diagrama de estado

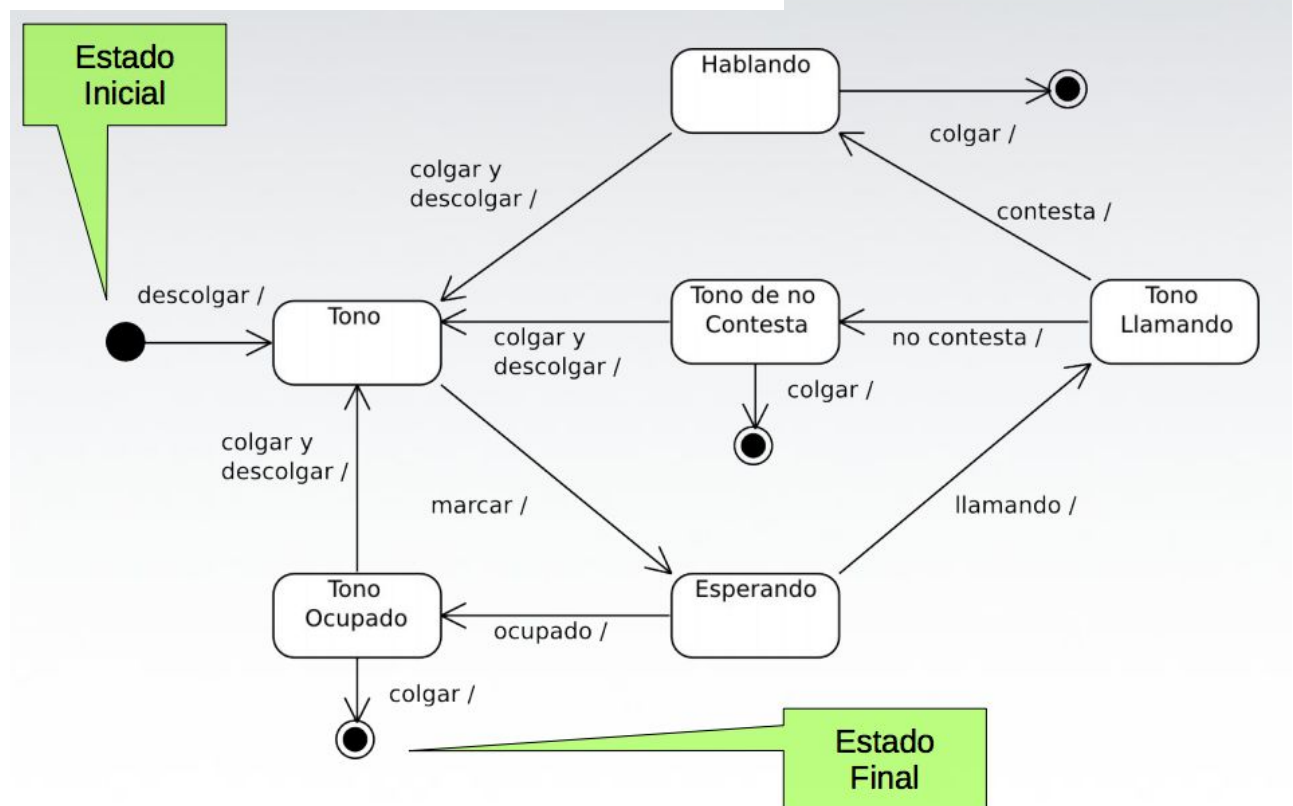


# Ejemplo

Diagrama de clases



Diagrama de estado

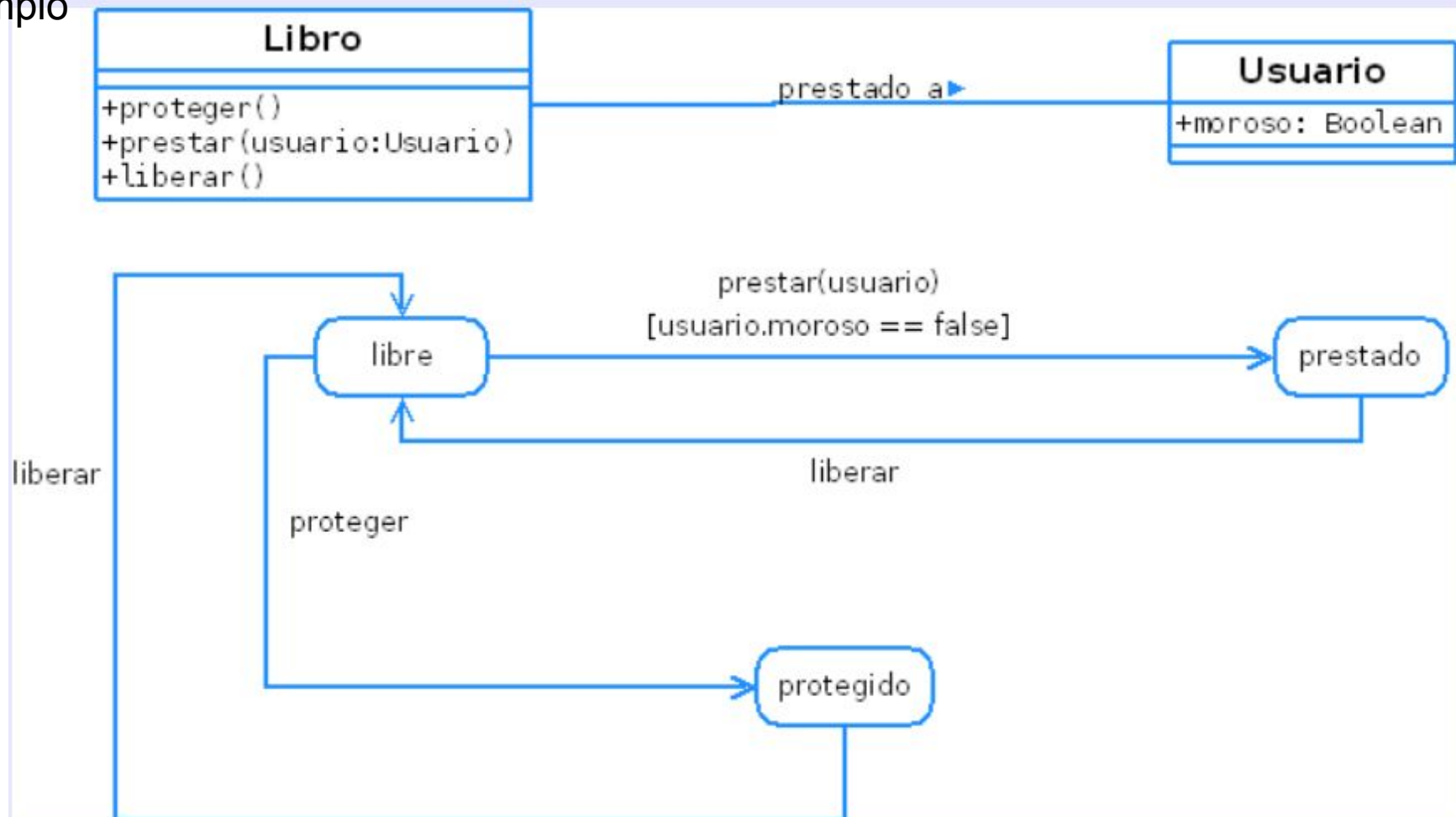




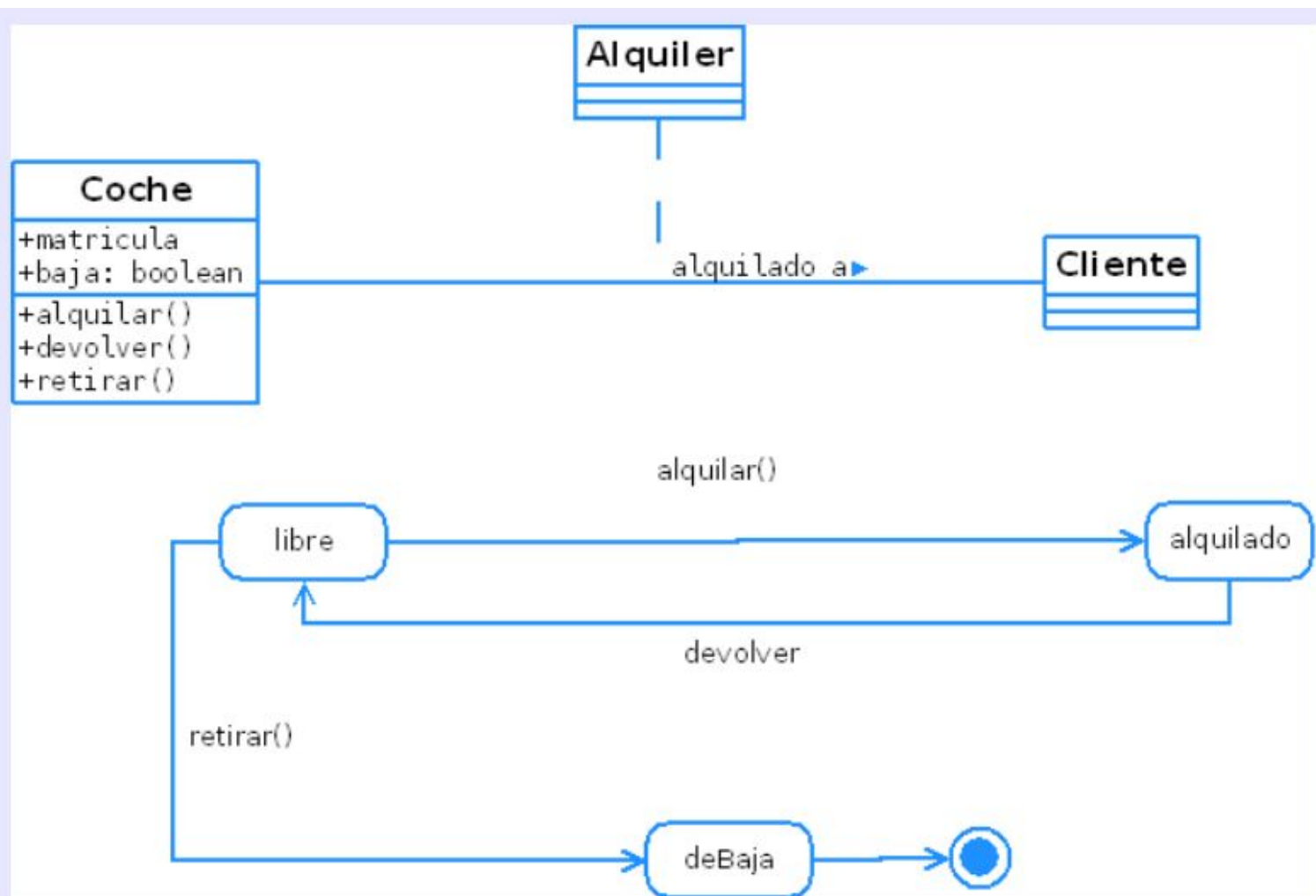
# Comparación con los diagramas de actividad

Un Diagrama de Actividades no es más que un caso especial de un diagrama de estados, en el que todos los estados (o la gran mayoría) son actividades

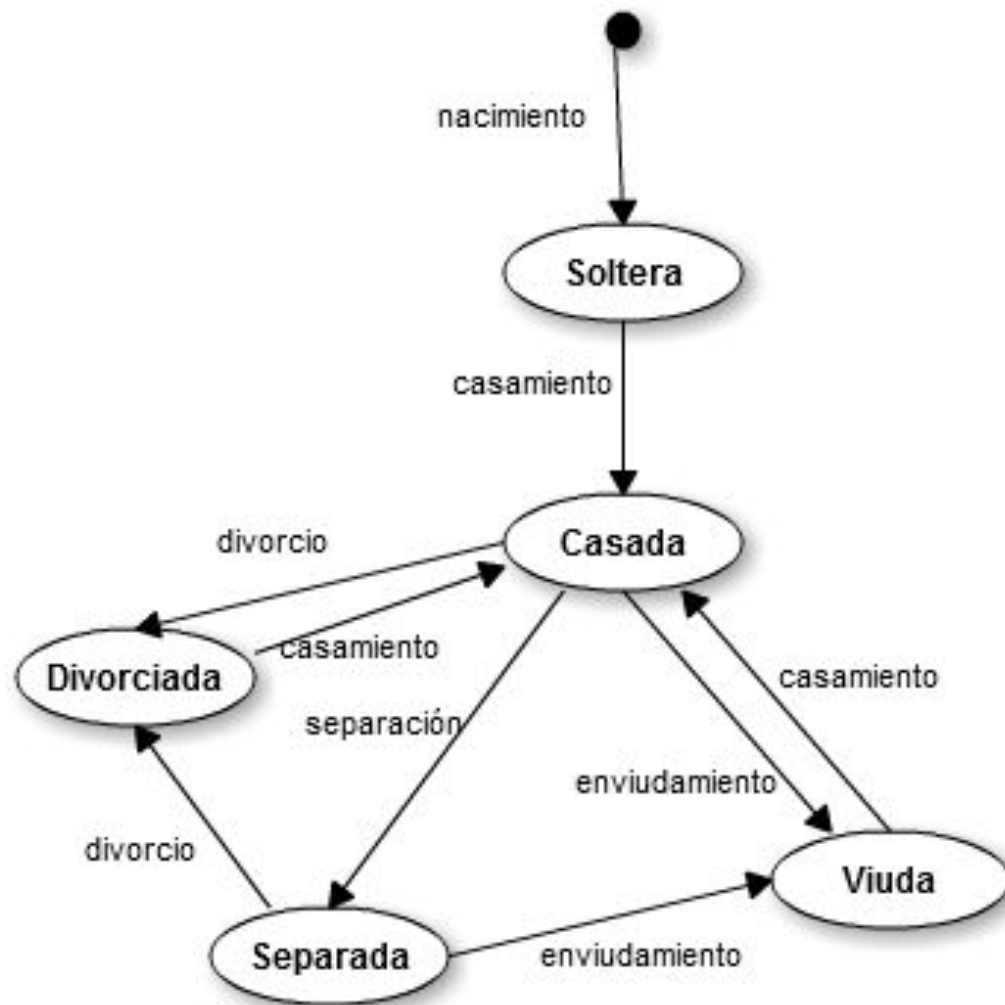
## Ejemplo



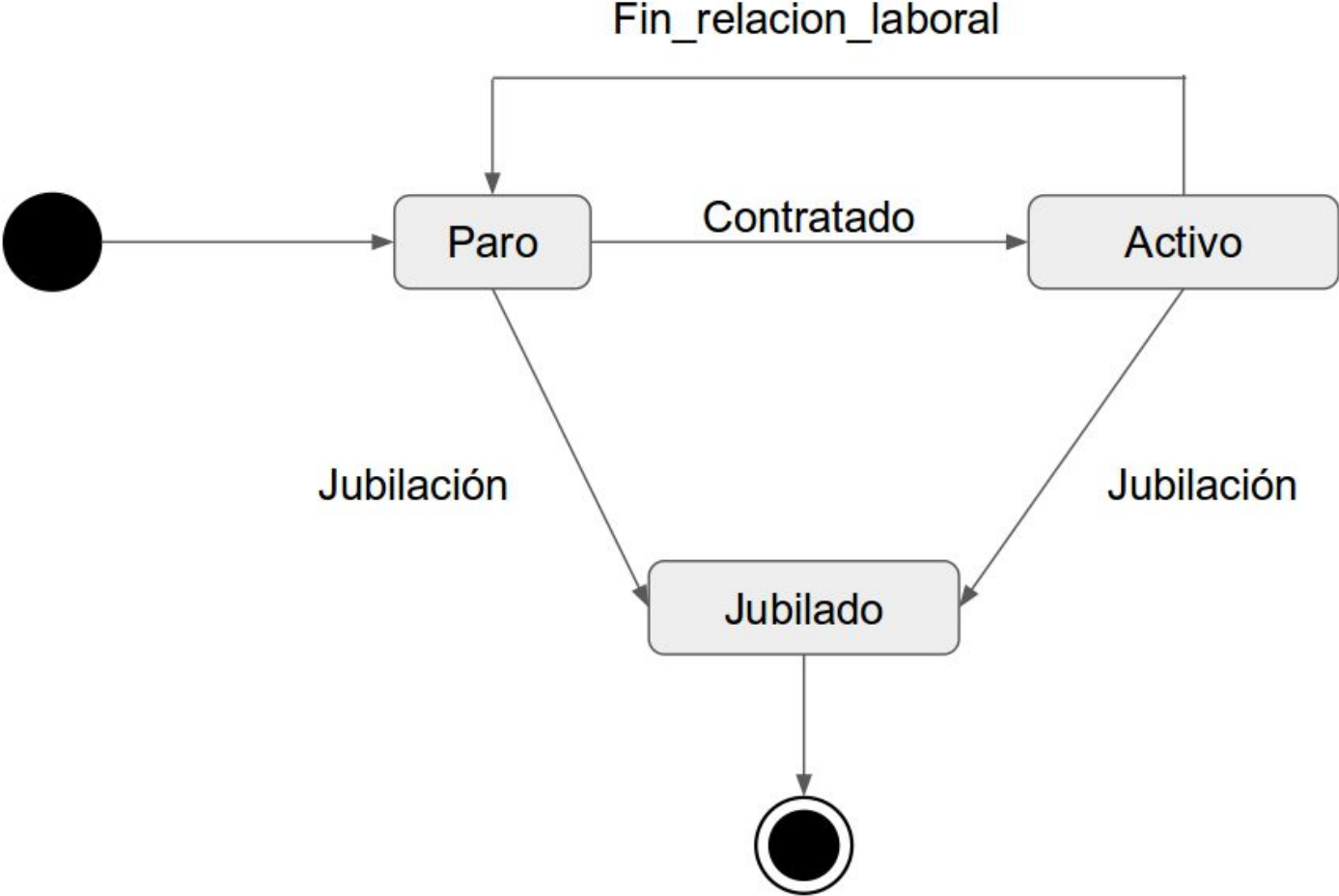
## Ejemplo



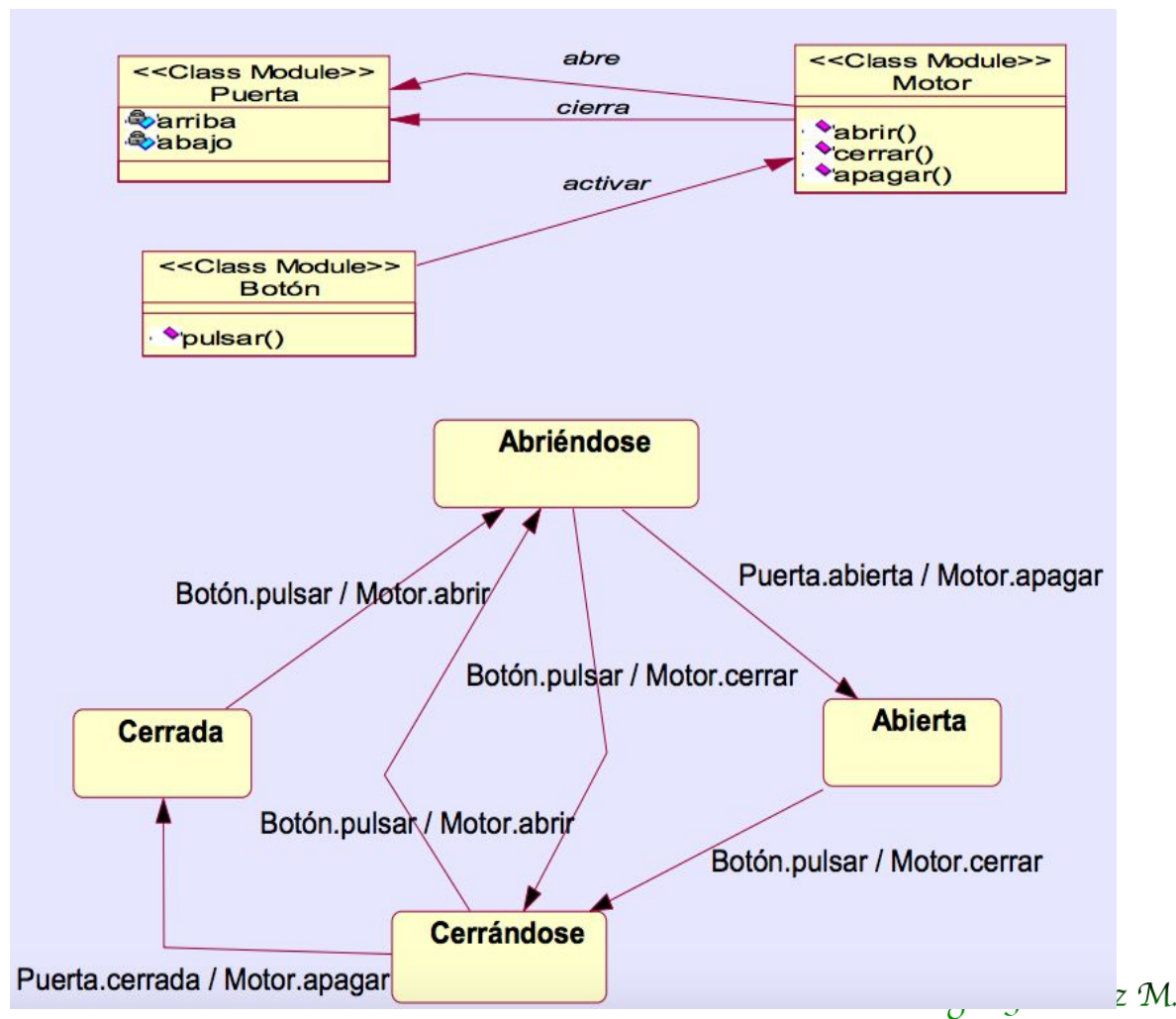
## Ejemplo



Ejemplo



# Ejemplo



# Ejemplo

