



Tema 5. Modelado de comportamiento del sistema

Modelado del Software

Escuela Politécnica



Contenidos

- Introducción
- Eventos
- Diagrama de Estados
- Diagrama de Actividades
- Conclusión



Contenidos


- **Introducción** ←
- Eventos
- Diagrama de Estados
- Diagrama de Actividades
- Conclusión



Modelando el comportamiento de un sistema

- En el tema anterior vimos las interacciones como mecanismo para modelar aspectos dinámicos de un sistema.
- En este tema estudiamos conceptos y diagramas empleados para modelar el **comportamiento dinámico** de un sistema determinado por **cosas que ocurren externa o internamente**.
- El comportamiento de un objeto individual se modela mediante una **máquina de estados**.
- Además de los diagramas de casos de uso, y de interacción (Secuencia y comunicación), para modelar aspectos dinámicos de un sistema también se usan:
 - **Diagramas de estados** (orientados a eventos)
 - **Diagramas de actividades** (orientados a actividades)

Contenidos

- Introducción
- **Eventos** 
- Diagrama de Estados
- Diagrama de Actividades
- Conclusión



Definición de evento

- **Evento:** *acontecimiento significativo, ubicado en el tiempo y en el espacio*
- Son los estímulos que hacen a una clase cambiar su estado
- Los eventos pueden ser:
 - **Externos:** fluyen entre el sistema y sus actores
 - **Internos:** fluyen entre objetos del sistema
- Pueden ser:
 - **Síncronos:** llamadas (invocación de operaciones)
 - **Asíncronos:** señales (excepciones), paso de tiempo, cambio de estado

Tipos de eventos (i)

- **Evento señal**

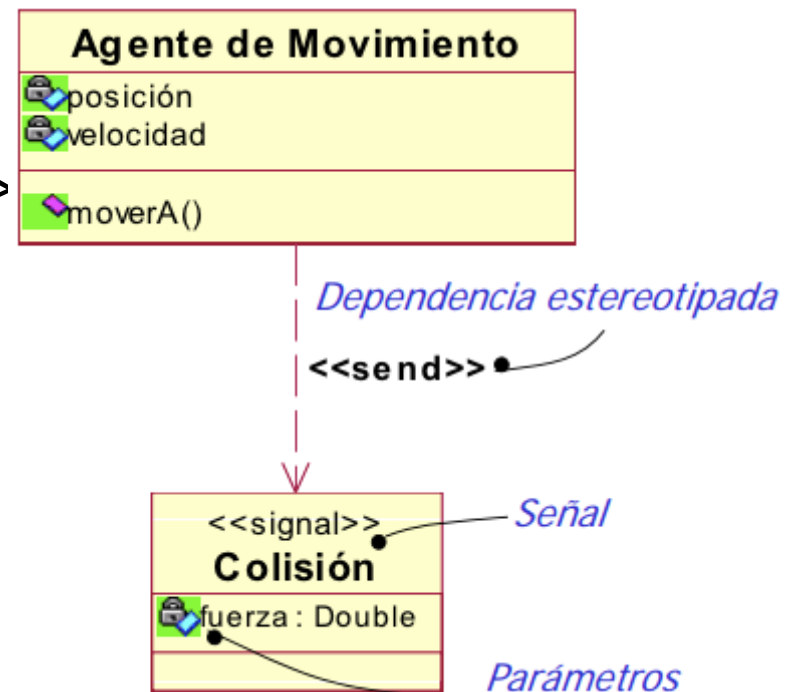
- Permite el **intercambio de objetos entre objetos**.
- Las señales son parecidas a las clases
 - Pueden tener instancias, atributos y operaciones
 - Pueden existir relaciones de generalización
- Una señal puede originarse por varios motivos:
 - Acción de transición de un estado a otro
 - Consecuencia de ejecutar una operación
 - Etc.



Tipos de eventos (ii)

- **Evento señal**

- Las señales se modelan como clases estereotipadas `<<signal>>`
- Para indicar que una operación envía una señal se usa la dependencia estereotipada `<<send>>`



Tipos de eventos (iii)

- **Evento llamada**

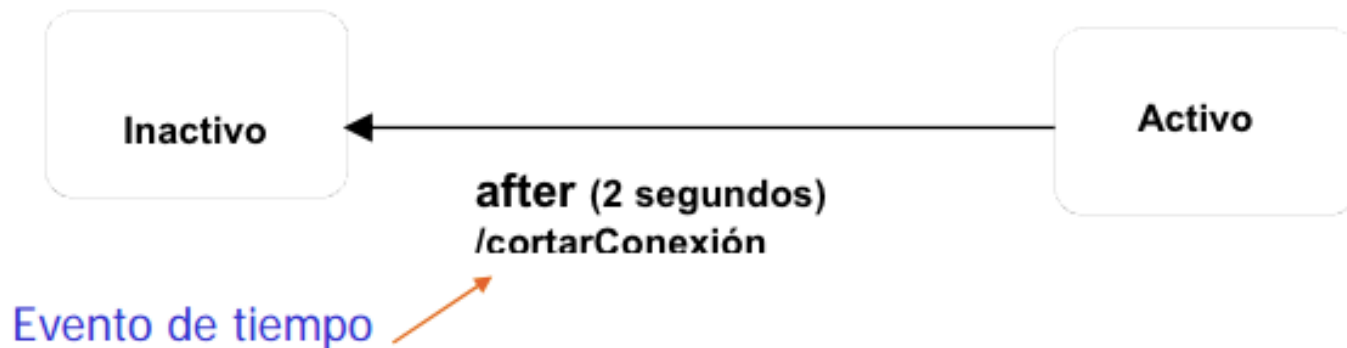
- Representa la invocación de una **operación de un objeto**
- Suele ser síncrono (esperar resultado de la operación)
- Dos casos especiales de evento llamada:
 - Creación de un objeto <<create>>
 - Destrucción de un objeto <<destroy>>
- Modelado de un evento llamada (similar a una señal):



Tipos de eventos (iv)

- **Evento de Tiempo**

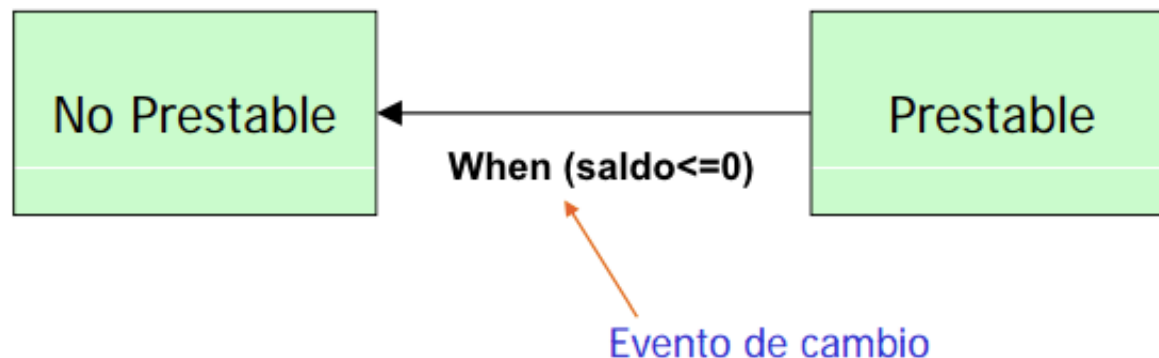
- Representa un **instante en el tiempo** mediante una expresión.
- La expresión puede ser:
 - **Absoluta:** **at** <valor temporal>
 - Ejm: at (21:00h)
 - **Relativa:** **after** <valor temporal>
 - Ejm: after (2 segundos)



Tipos de eventos (v)

- **Evento de Cambio**

- Representa un **cambio** de estado o el cumplimiento de alguna **condición**.
- Notación: **When (<expresion booleana>)**
- Ocurre una vez cuando el valor de la expresión cambia de **falso -> verdadero**
 - El evento no ocurre si la expresión pasa de verdad -> falso
 - No se repite mientras la expresión siga siendo verdadera.

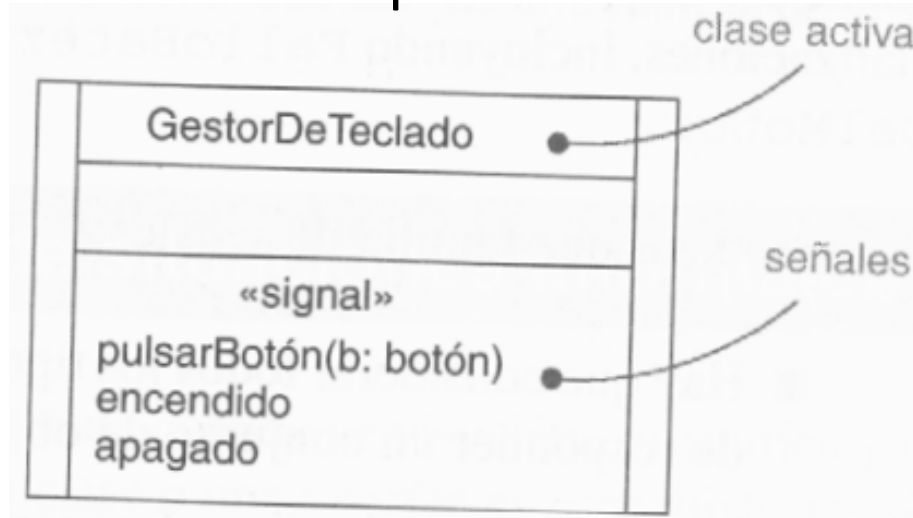


Uso de los tipos de eventos

- Eventos de tipo **cambio** o **tiempo** son de **uso local** en un objeto.
- Notar que los eventos **señal** o **llamada** implica **al menos dos objetos**:
 - Emisor: el que envía la señal o invoca la operación
 - Receptor: el que recibe la señal o implementa la operación
- Cualquier objeto puede:
 - Enviar eventos de tipo señal a otros objetos
 - Invocar operaciones (i.e. enviar eventos de tipo llamada) a otros objetos
 - Recibir eventos de tipo señal o llamada

Diagrama de Estados

- Uso de los tipos de eventos
 - Los eventos de **llamada** que puede recibir un objeto se modelan como **operaciones/métodos** sobre la clase objeto.
 - Las **señales** que puede recibir un objeto se modelan designándolas en un compartimento extra de la clase.



Contenidos


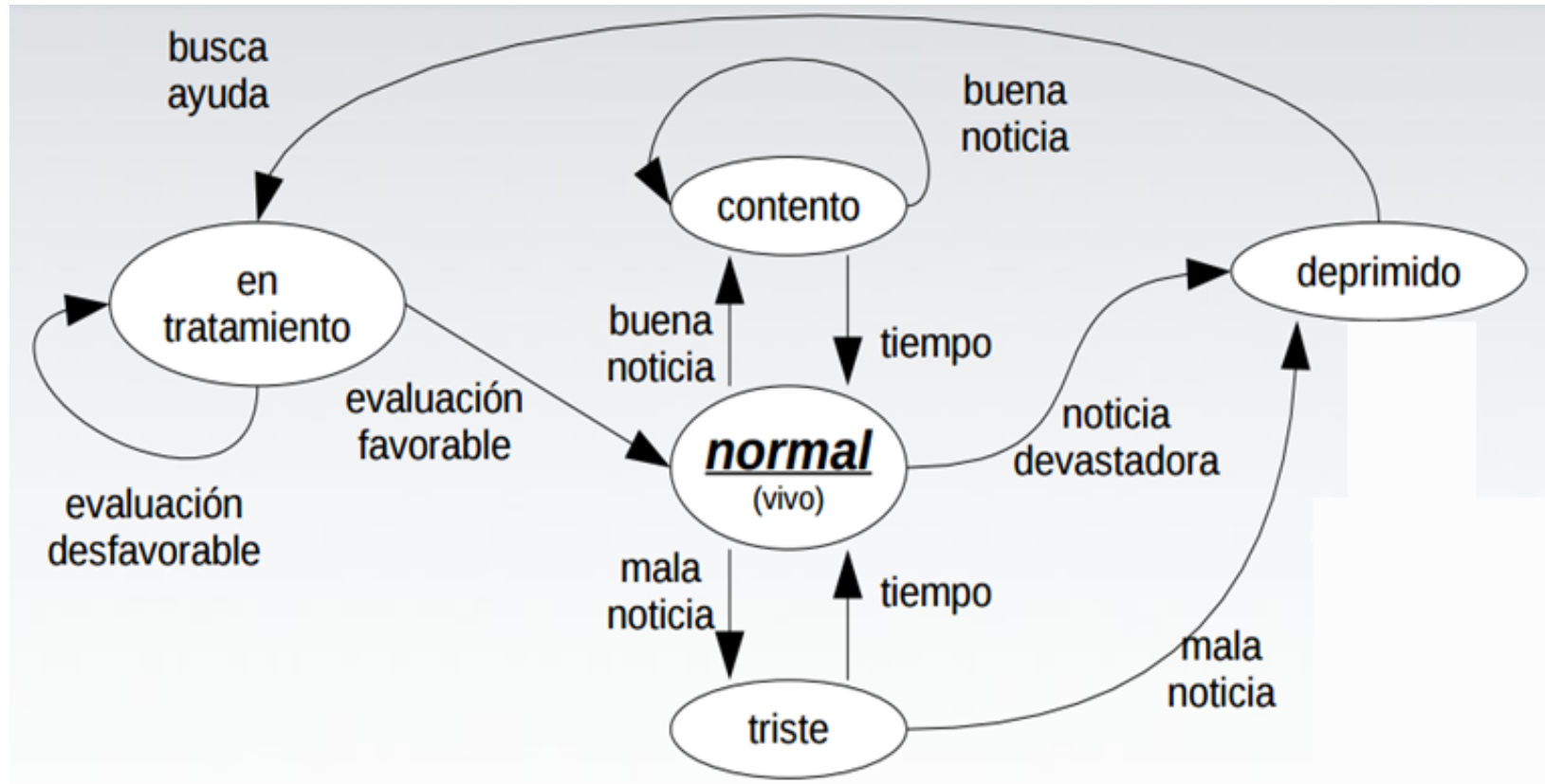
- Introducción
- Eventos
- **Diagrama de Estados** 
- Diagrama de Actividades
- Conclusión



Diagrama de Estados

- Muestra **máquinas de estados**
- Consta de **estados, transiciones, eventos y acciones**
 - **Estado**: situación de un objeto.
 - **Evento**: acontecimiento que ocurre en un instante concreto.
 - **Transición**: relación entre dos estados. Al pasar de uno a otro se realizan acciones. Para entrar el segundo estado se indica condiciones.
 - **Acción**: actividad o tarea que realiza el sistema, provocando un cambio de estado.
- Cubren la **vista dinámica de un objeto**.
- Resaltan el comportamiento dirigido por eventos de un objeto.

Diagrama de Estados



Definición de máquina de estados

- ➡ Secuencia de **estados**
- ➡ Por las que pasa un **objeto** a lo largo de su **vida**
- ➡ Como consecuencia de **procesar eventos** que recibe

-Notar que...

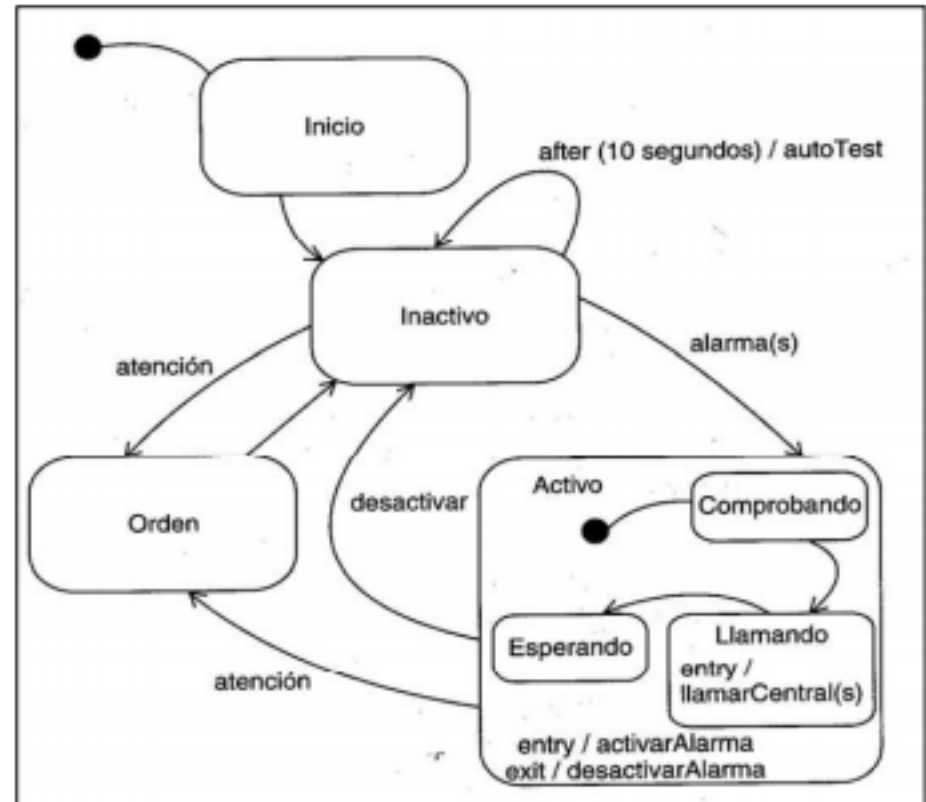
- Modelan el **comportamiento de un objeto**.
- El comportamiento de una sociedad de objetos lo modelan los diagramas de interacciones.

Uso de máquinas de estados

- Son la mejor forma de especificar el comportamiento de los **objetos reactivos**
 - Objetos que responden ante eventos externos o internos
 - Tienen “**memoria**”, es decir, su comportamiento depende de la historia
 - Tiene un ciclo de vida bien definido, con un progresión basada en estados y transiciones entre éstos en base a eventos.
- Una máquina de estados puede visualizarse de dos formas:
 - Destacando los estados de los objetos y sus transiciones (**Diagramas de estados**)
 - Destacando el flujo de control entre actividades (**Diagramas de Actividades**)

Máquina de estados (i)

- Una máquina de estados tiene forma de grafo dirigido con diferentes tipos de arcos y nodos
 - **Nodos**: representan los distintos estados por los que pasa un objeto
 - **Arcos**: representan las transiciones entre dichos estados. Normalmente estas transiciones son disparados por **eventos**.



Máquina de estados (ii)

- **Estado:**

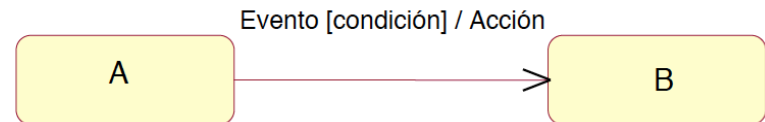
- Condición o situación en la vida de un objeto
- Un objeto permanece en un estado un tiempo finito de tiempo
- En un estado, el objeto hace alguna actividad o espera algún evento.

- **Transición:**

- Relación entre dos estados de un objeto
- Se transita entre entre estados cuando ocurra un evento y se satisfagan algunas condiciones.

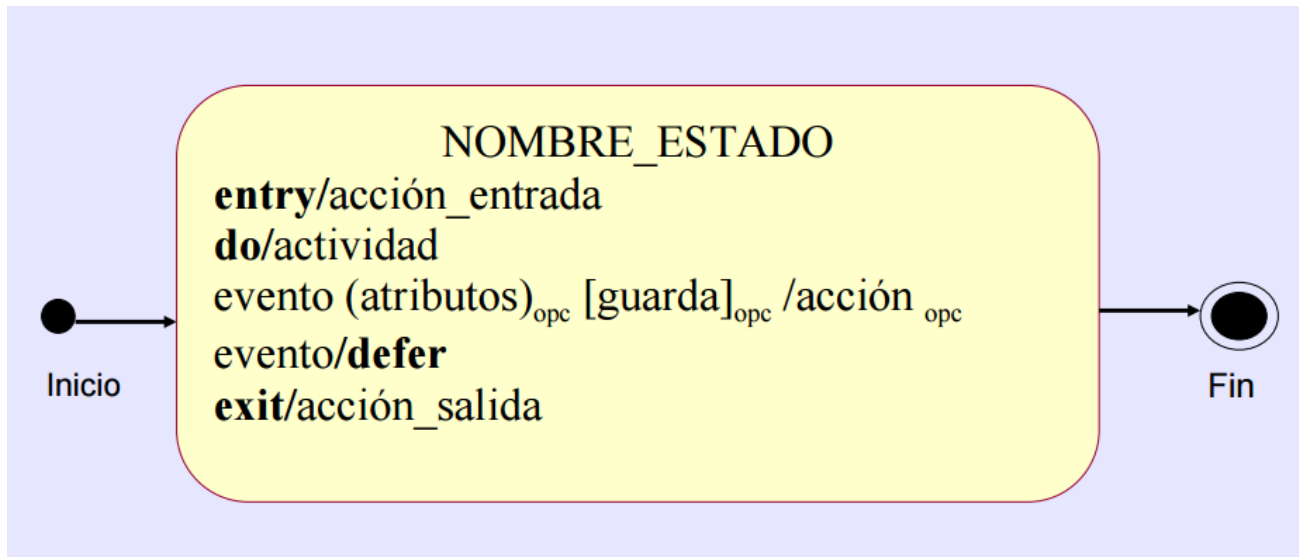
- Componentes:

- **Estado origen / destino**
- **Evento disparo:** provoca el disparo de la transición
- **Condición de guarda:** expresión booleana. Sólo si es verdadera se puede disparar la transición.
- **Acción:** Comportamiento ejecutable que realiza el objeto.



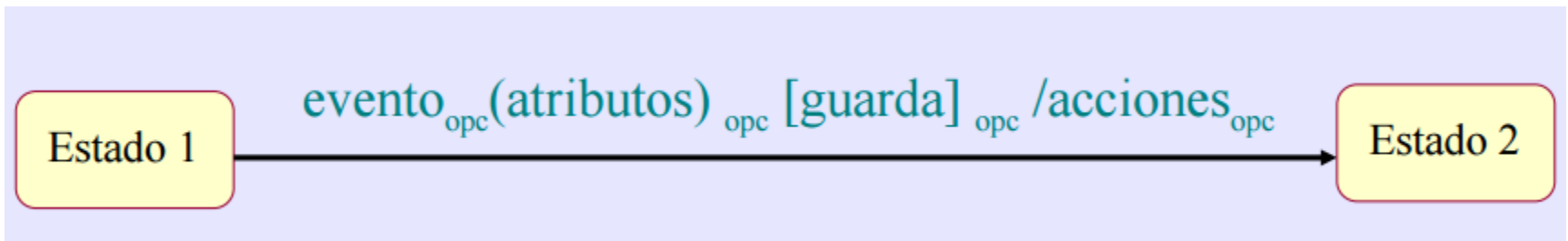
Máquina de estados (ii)

- Estado:

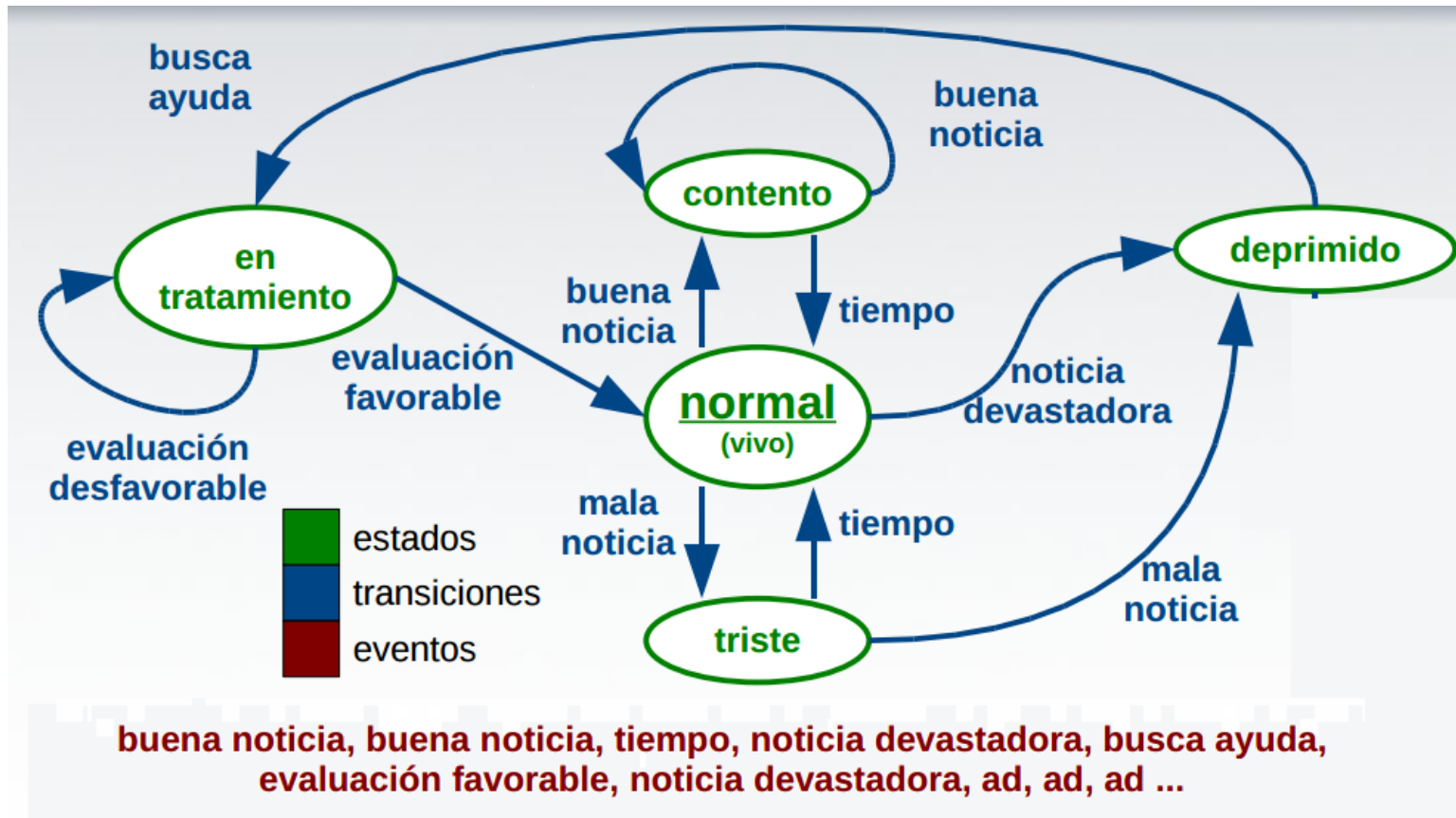


Máquina de estados (ii)

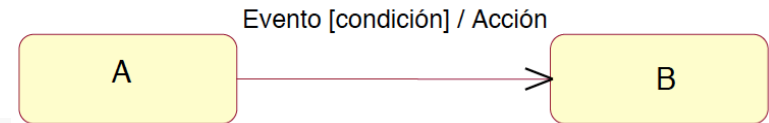
- **Transición:** La transición puede tener tres elementos (opcionales): evento [guarda] / acción.



Máquina de estados (ii)



Máquina de estados (iii)



**ESTADO
INICIAL**

**ESTADO
FINAL**

Diagrama de Estados

- Es una máquina de estados de un objeto
- Gráficamente:
 - **Grafo dirigido**: colección de nodos y arcos
 - **Estado inicial y final** claramente diferenciados del resto



Diagrama de Estados

- Ejemplo de diagrama de estados de objetos de la clase “Socio”:

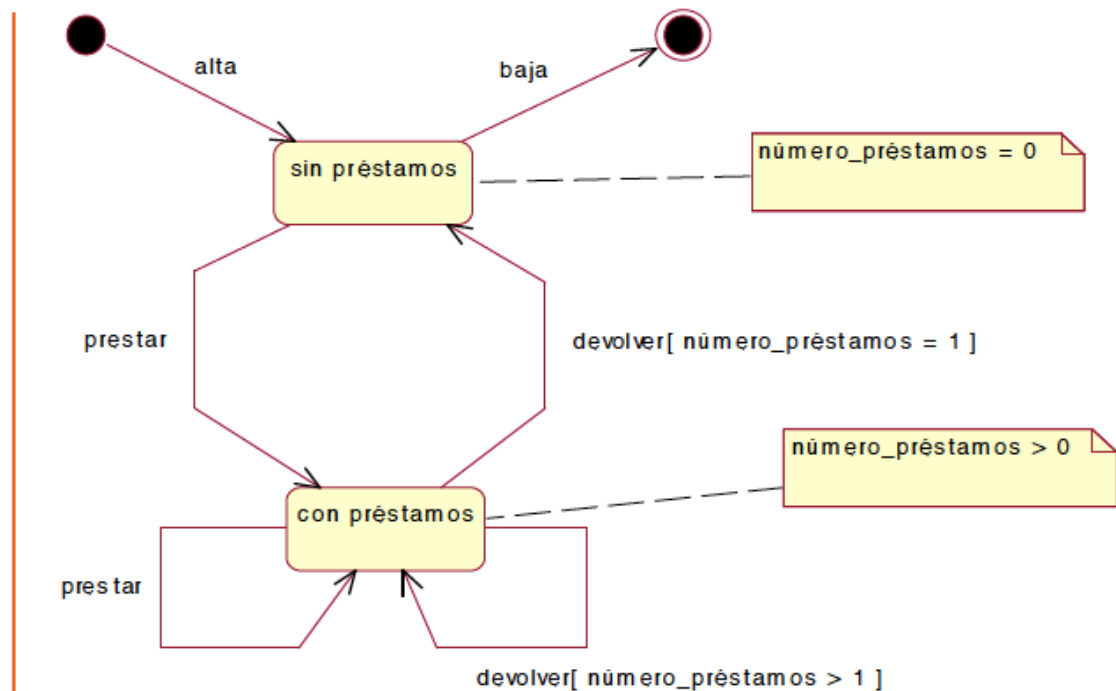
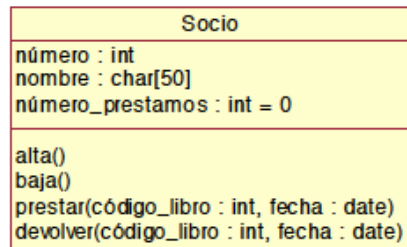


Diagrama de Estados

- Ejemplo de diagrama de estados de objetos de la clase “Persona”:

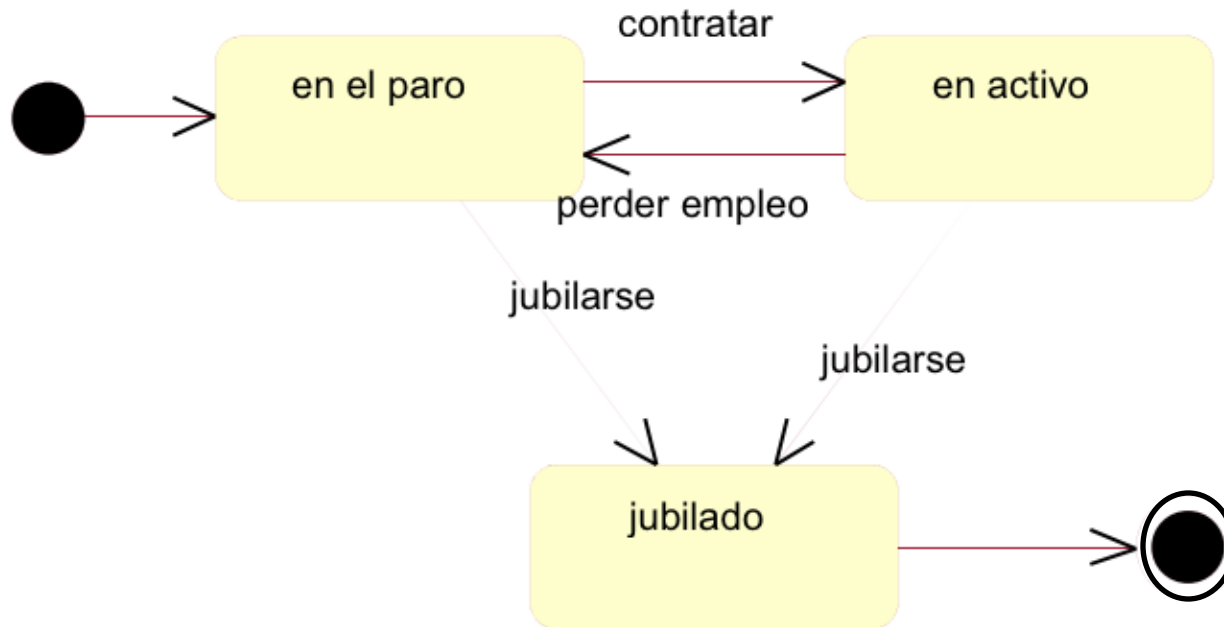


Diagrama de Estados

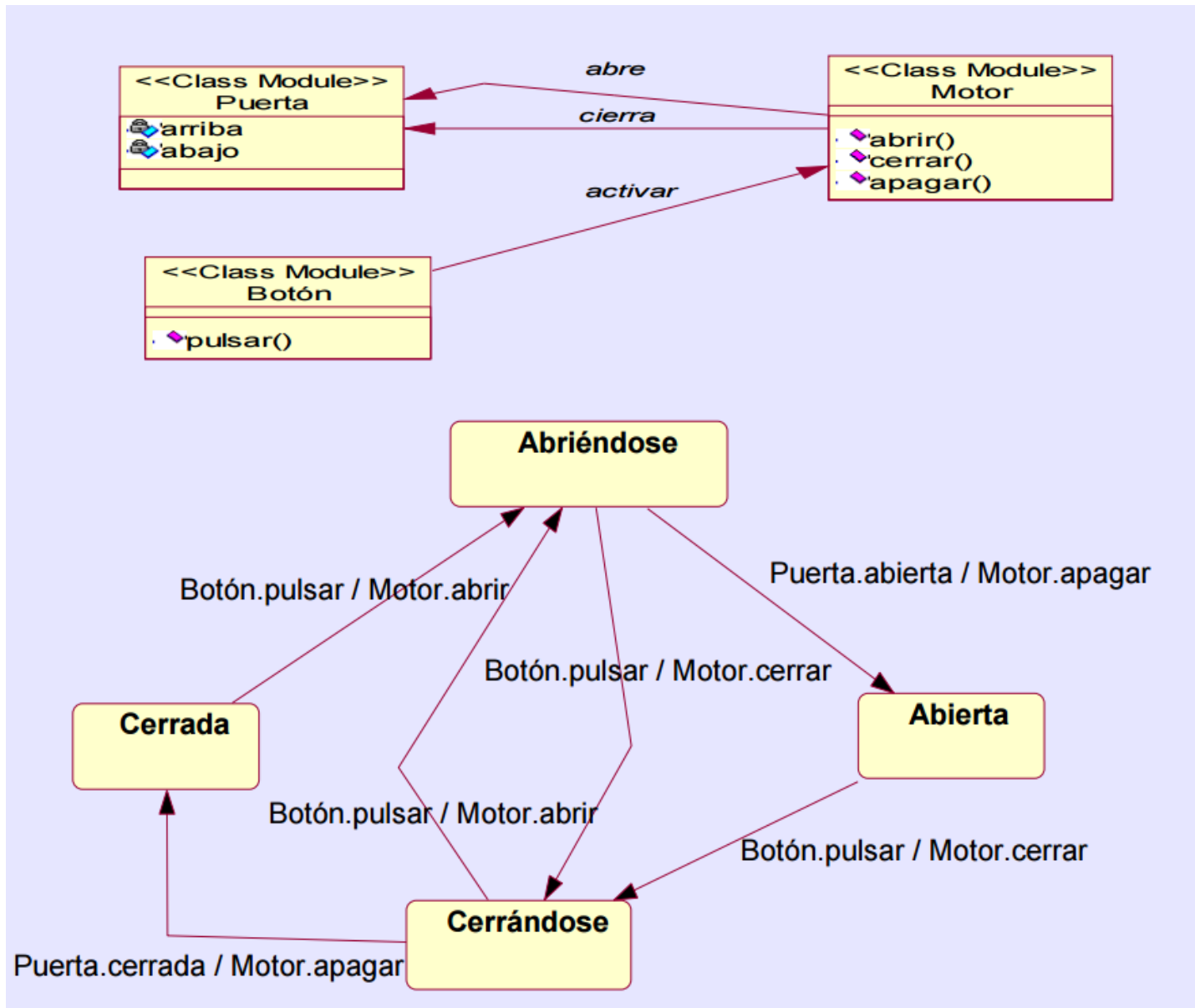


Diagrama de Estados

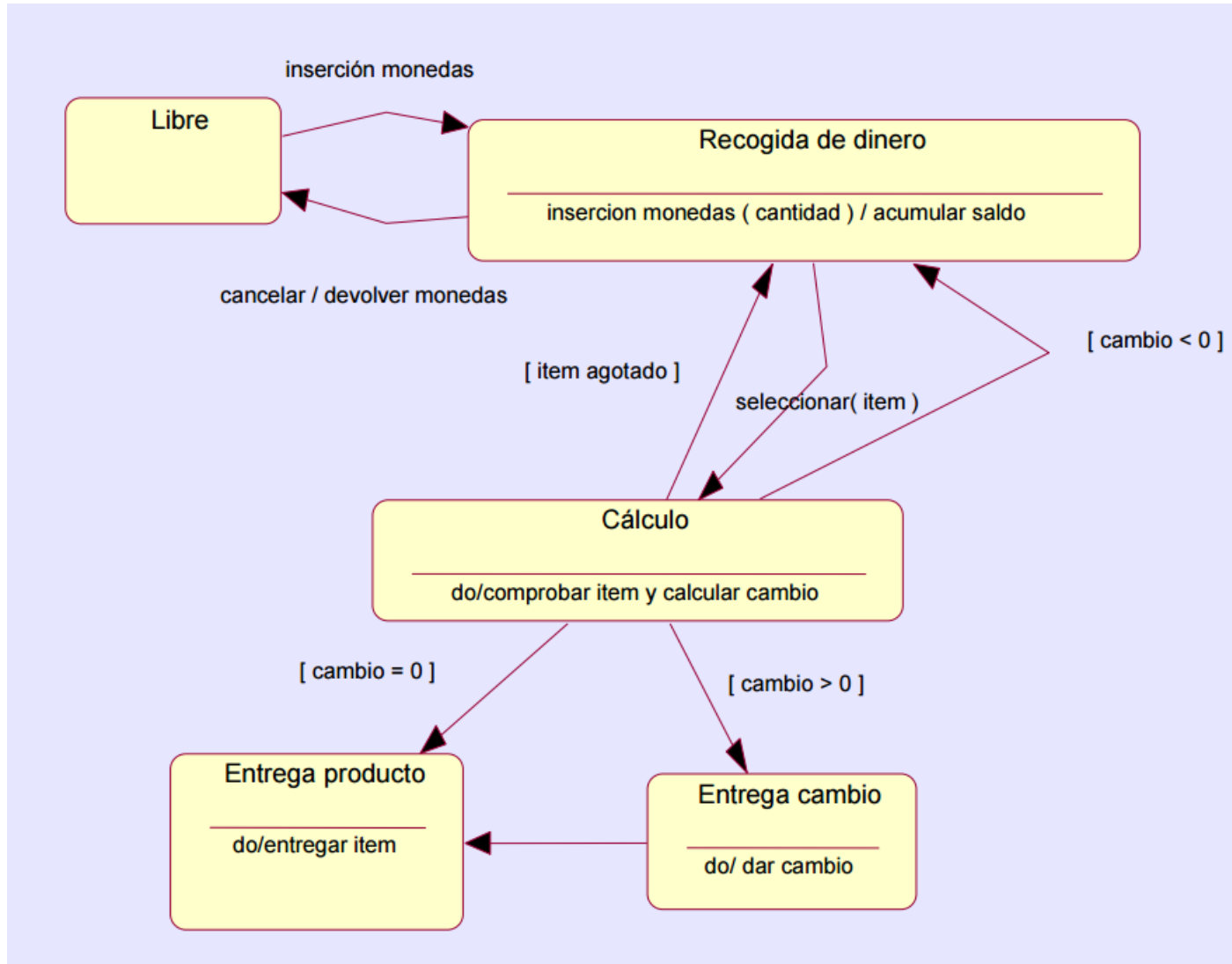


Diagrama de Estados

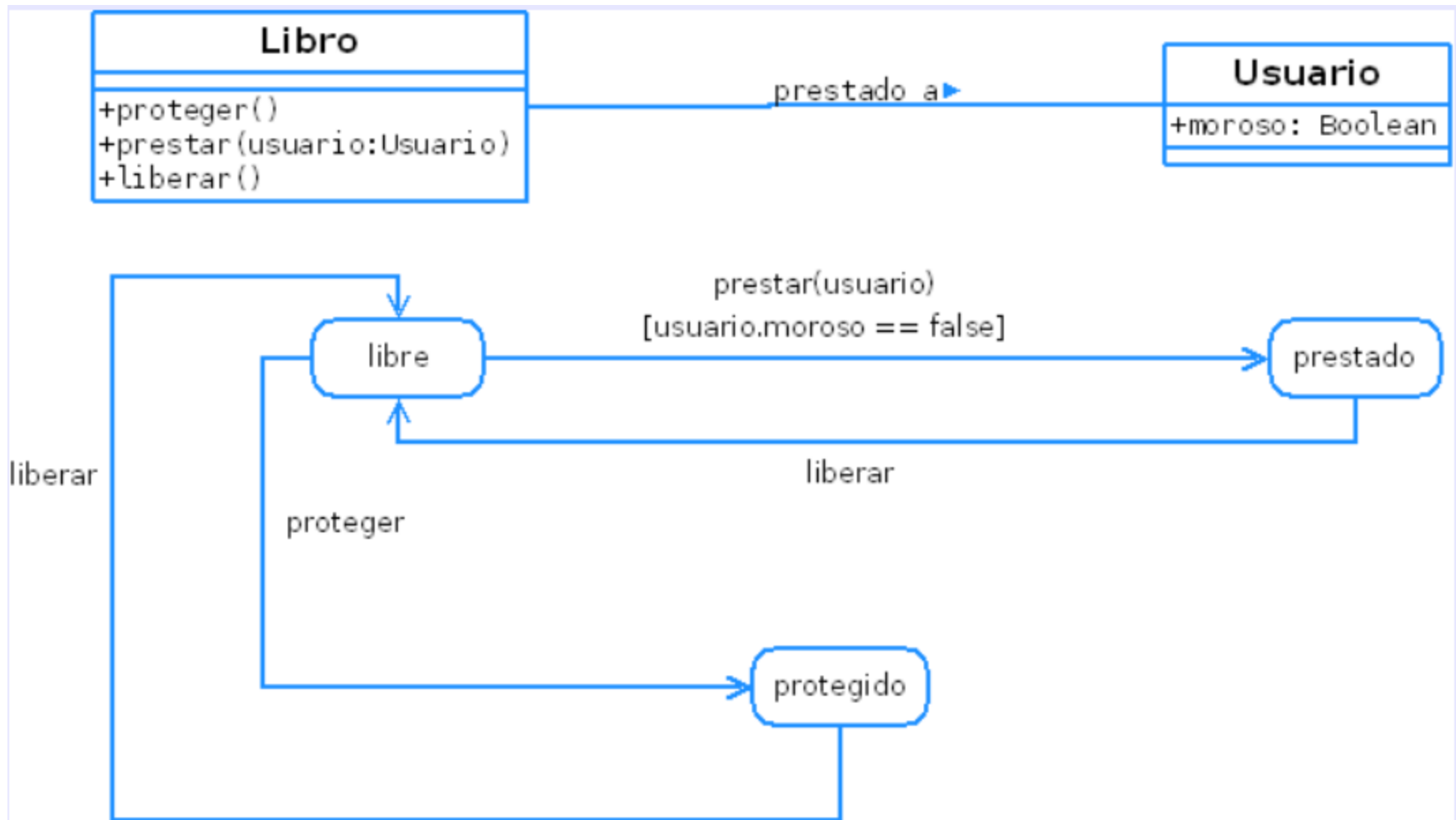


Diagrama de Estados

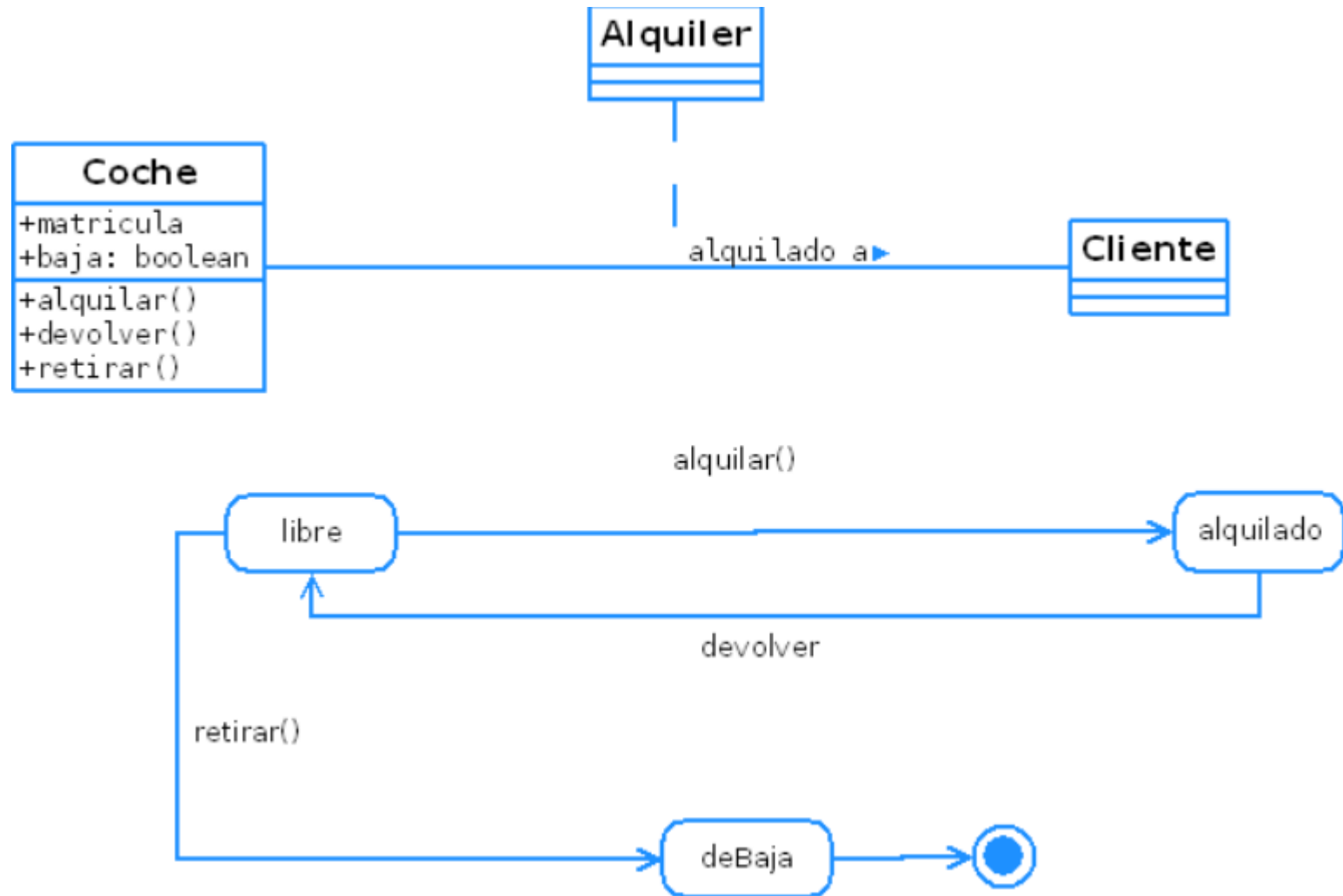


Diagrama de Estados

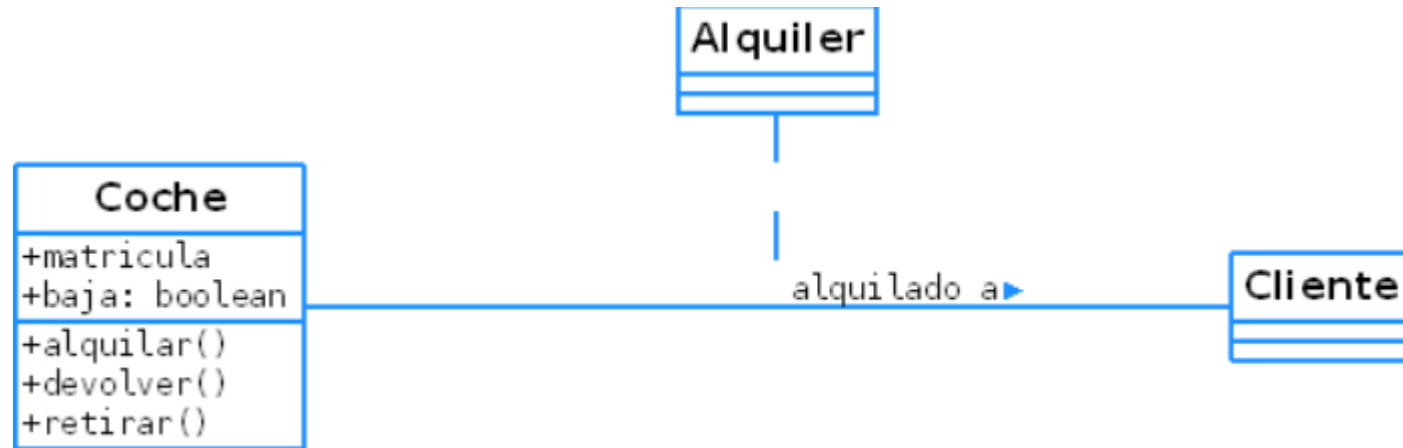
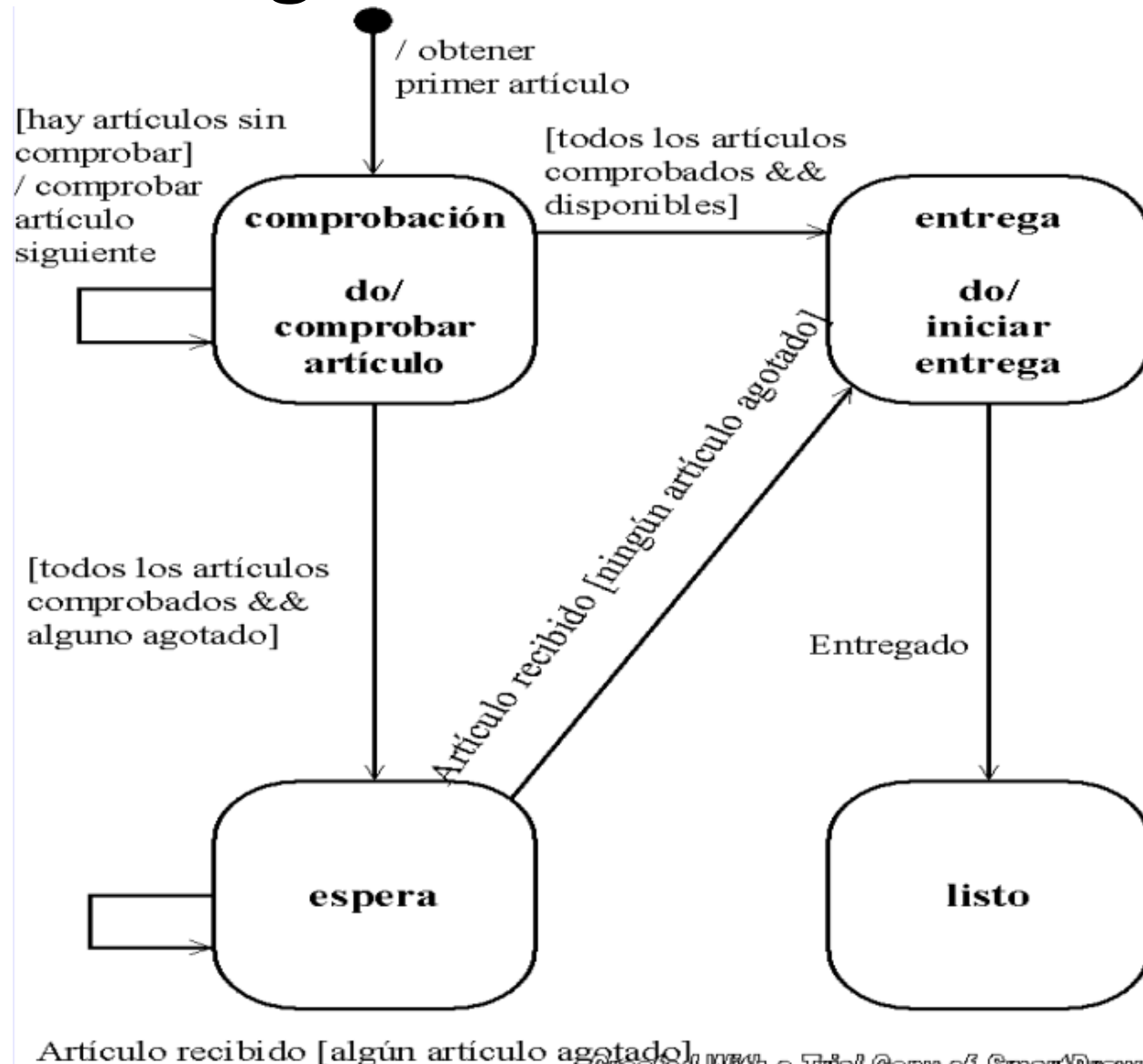


Diagrama de Estados



Artículo recibido [algun artículo agotado]

Contenidos


- Introducción
- Eventos
- Diagrama de Estados
- **Diagrama de Actividades** 
- Conclusión



Diagrama de Actividades

- Muestran el flujo de trabajo paso a paso de una computación que implica la coordinación de varios objetos.
- Cubren la **vista dinámica de un sistema**
- Son el **equivalente** en OO a los **diagramas de flujo y DFDs**.
- Se emplean para especificar el **proceso de negocio** o **flujo de trabajo asociado a un caso de uso**.

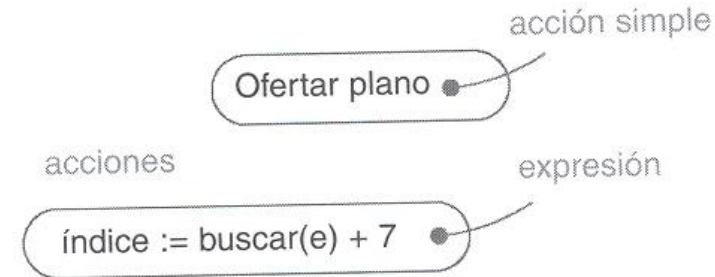
Actividad

- Es una ejecución no atómica dentro de una máquina de estados.
- Una actividad consiste en un **flujo** entre **nodos de actividad** que **produce/consume información** valores.
- Nodo de actividad = ejecución de un conjunto de **acciones**
 - Puede suceder que *Nodo de actividad = 1 acción*

Elementos de un diagrama de actividad

- **ACCIÓN**

- Computación ejecutable y atómica
- No se descompone ni ejecuta por partes.
- Ejemplos: llamadas a otras operaciones, envío de señales, creación o destrucción de objetos, cálculos, etc.

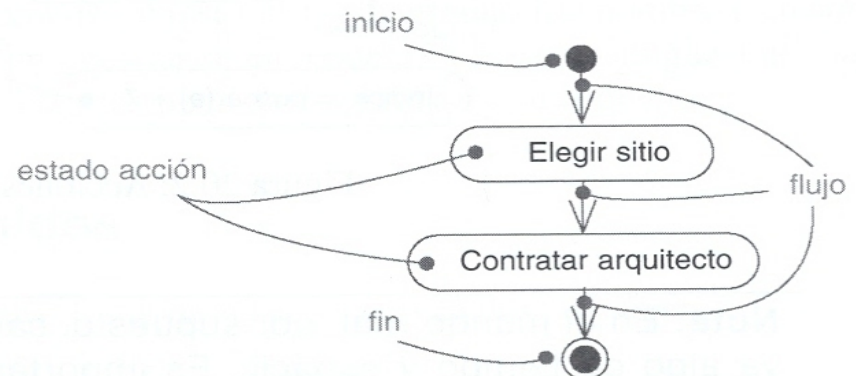


- **NODO DE ACTIVIDAD**

- Agrupación de acciones
- **No existe diferenciación gráfica** entre acciones y nodos de actividad → Usar anotaciones

- **FLUJO**

- Para marcar los pasos de una acción o nodo a otra.



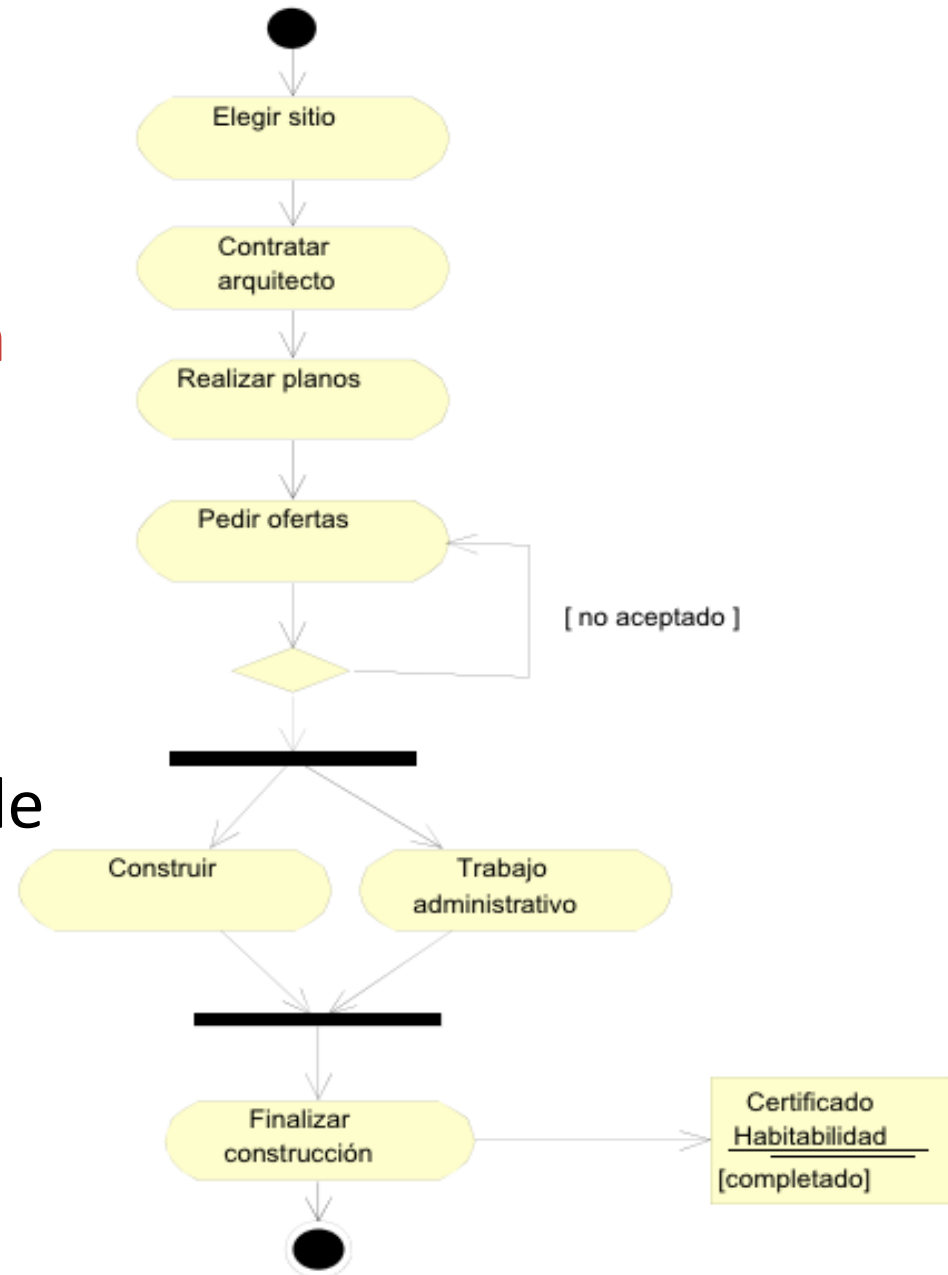
Componentes de un diagrama de actividad

- **Acciones:** nodos de actividad atómicos
- **Actividades:** nodos de actividad con estructura interna
- **Datos utilizados**
- **Nodos de Control:** controlan el flujo

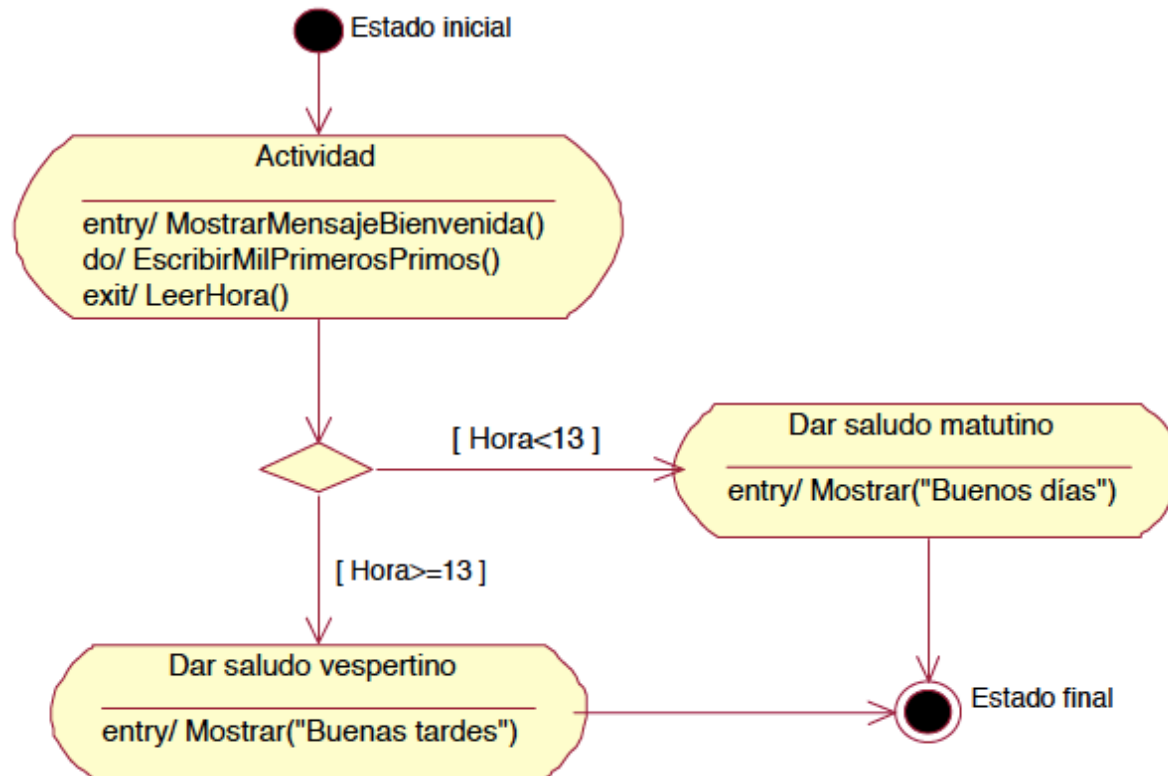


Diagrama de Actividades

- Debe haber **un inicio y un fin de actividad**
- Cuando se completa una acción o un nodo de actividad, se pasa a la siguiente acción o nodo de actividad.
 - Se usa una **flecha**



Ejemplo



Tipos de flujos

- Los **flujos secuenciales** son los más usados, pero existen otros flujos alternativos.
- Tipos de flujos:
 - **Bifurcación**: caminos alternativos en función de alguna condición
 - **División y Unión**: flujos de ejecución concurrentes
 - **Calles**: identificar responsables de cada actividad

Tipos de flujos (i)

- **Bifurcación**



- Caminos **alternativos** en función de una expresión booleana
- Gráficamente se representa como un rombo.
- Tiene un flujo de entrada y dos o más de salida
- Cada flujo de salida tiene una guarda.
 - Deben cubrir todas las posibilidades
 - Puede usarse “else” para marcar flujo de salida alternativo.

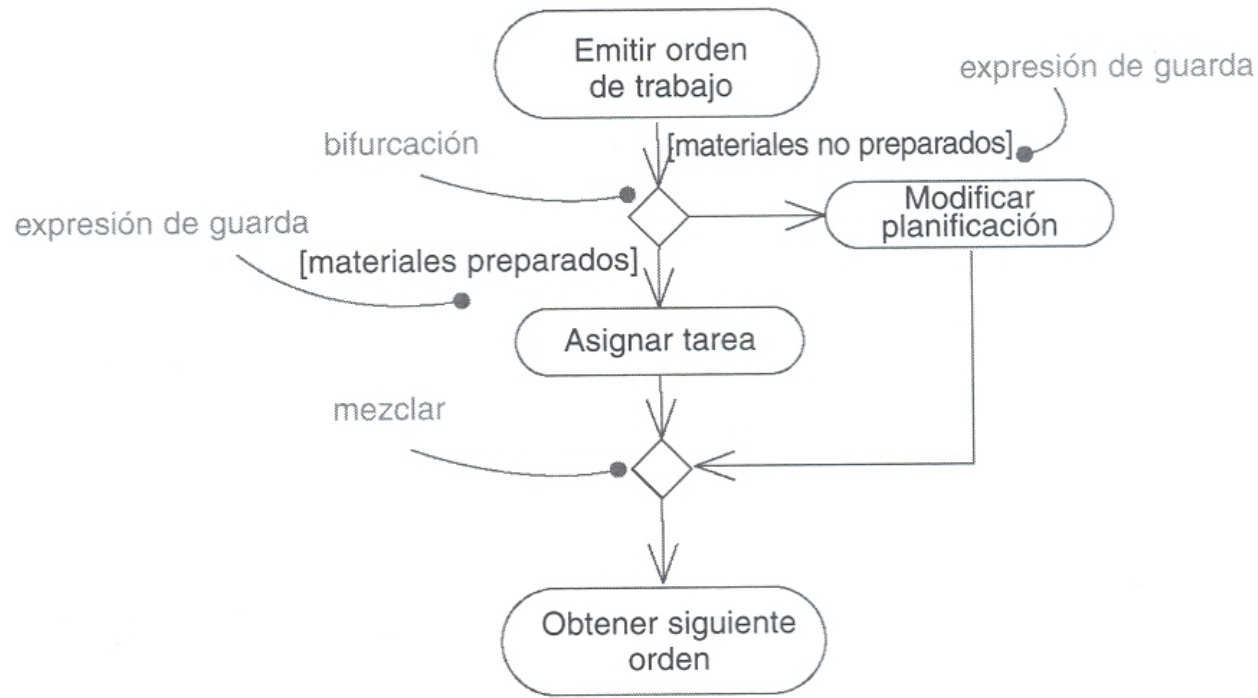
- **Fusión**



- Los caminos antes separados se vuelven a juntar en un rombo
- Tiene varias entradas y una única salida.
- Aquí no hay guardas

Tipos de flujos (ii)

- Ejemplo de bifurcación y fusión



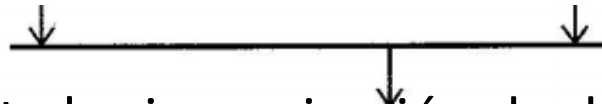
Tipos de flujos (iii)

- **División**



- Representa la separación de un flujo de control sencillo en dos o más flujos de control **concurrentes**.
- Tiene una transición de entrada y dos o más de salida.
- Después de la división, las actividades de cada camino continúan en paralelo.

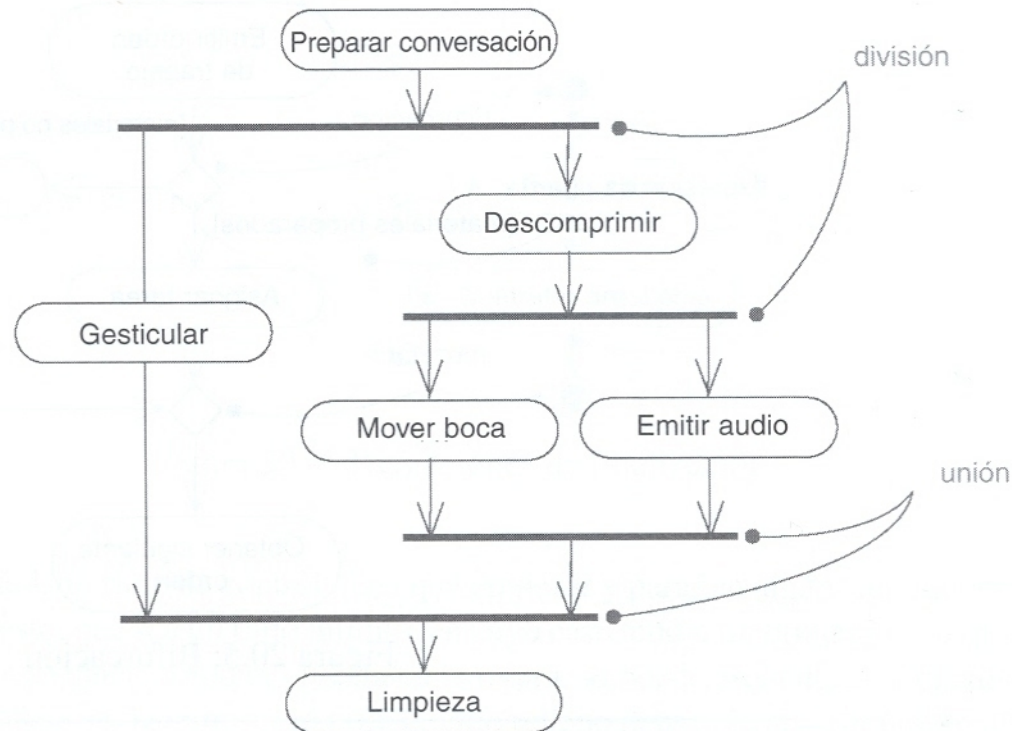
- **Unión**



- Representa la sincronización de dos o más flujos de control concurrentes.
- Tiene dos o más transiciones de entrada y una de salida.
- El flujo continúa cuando todos los flujos de entrada han alcanzado la unión.

Tipos de flujos (iv)

- Ejemplo de división y unión



Calles (i)

- Son agrupaciones de flujos de procesos
- Las transiciones pueden cruzarlas
- Cada calle representa una responsabilidad de alto nivel de la actividad que se está modelando.



Calles (ii)

- Ejemplo con calles

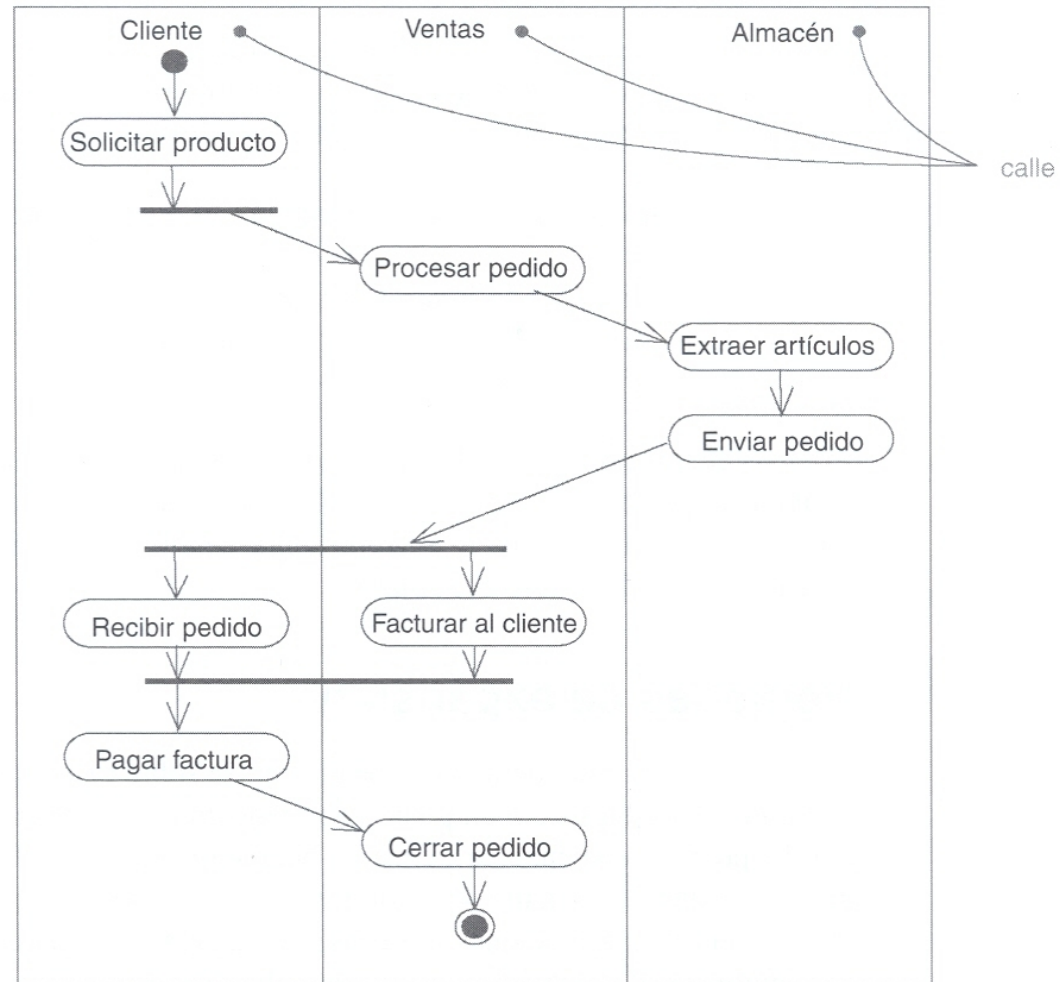


Diagrama de Actividades

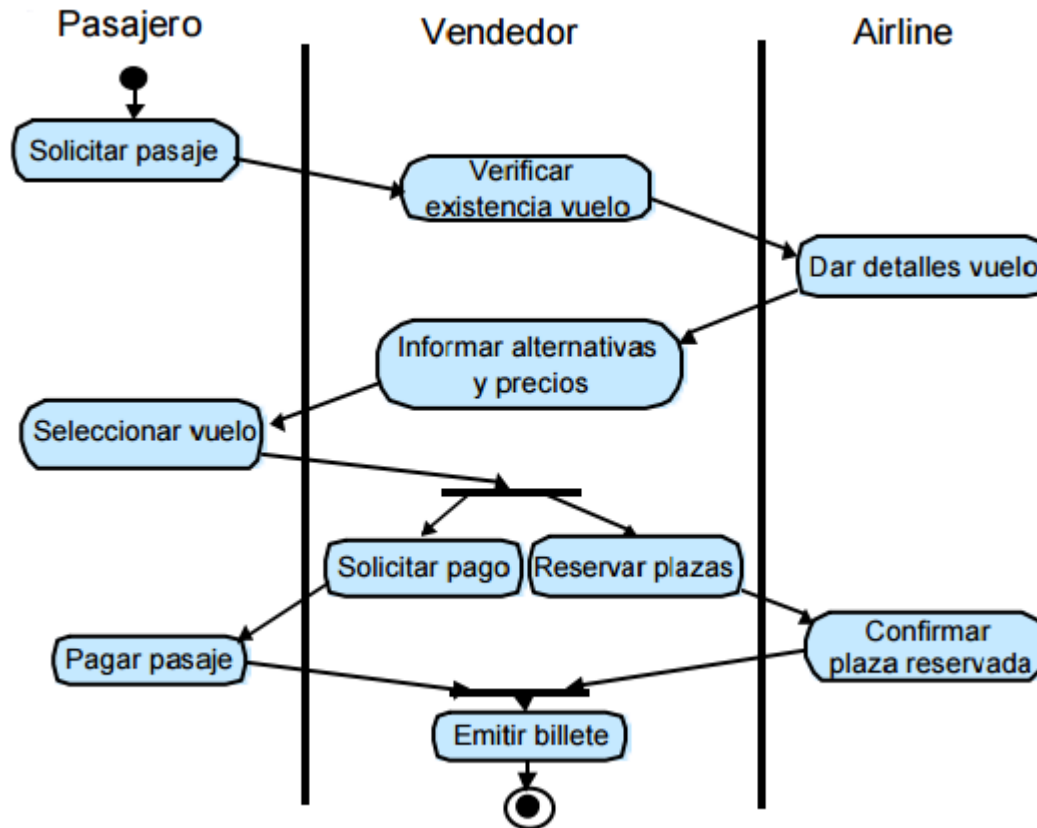
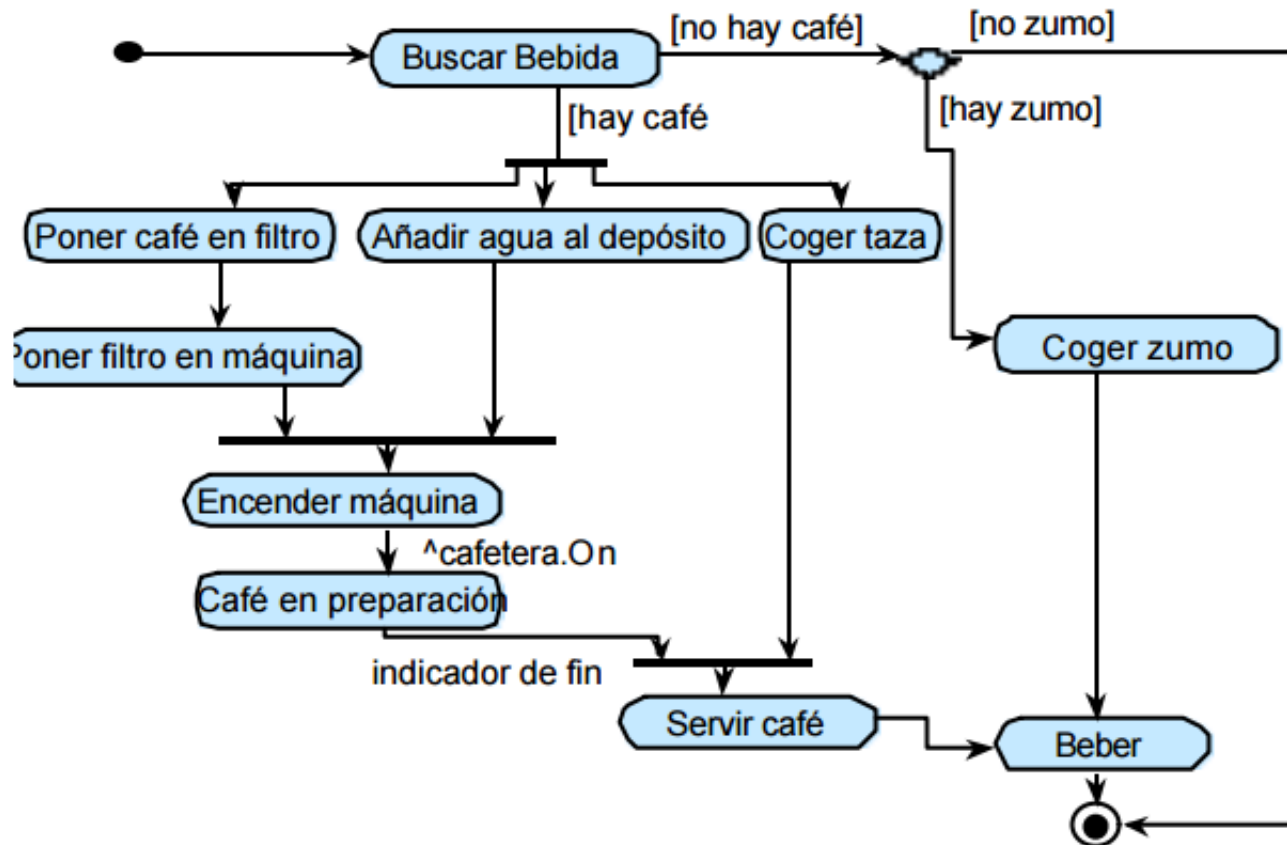


Diagrama de Actividades



Contenidos


- Introducción
- Eventos
- Diagrama de Estados
- Diagrama de Actividades
- **Conclusión** 



Diagrama de estados **vs** Diagrama de Actividades

- Ambos tipos de diagramas permiten modelar aspectos de comportamiento dinámicos, en el contexto del sistema global

Pero...¿cuándo usar uno u otro?



Diagrama de estados **vs** Diagrama de Actividades

- **Diagrama de Estados**

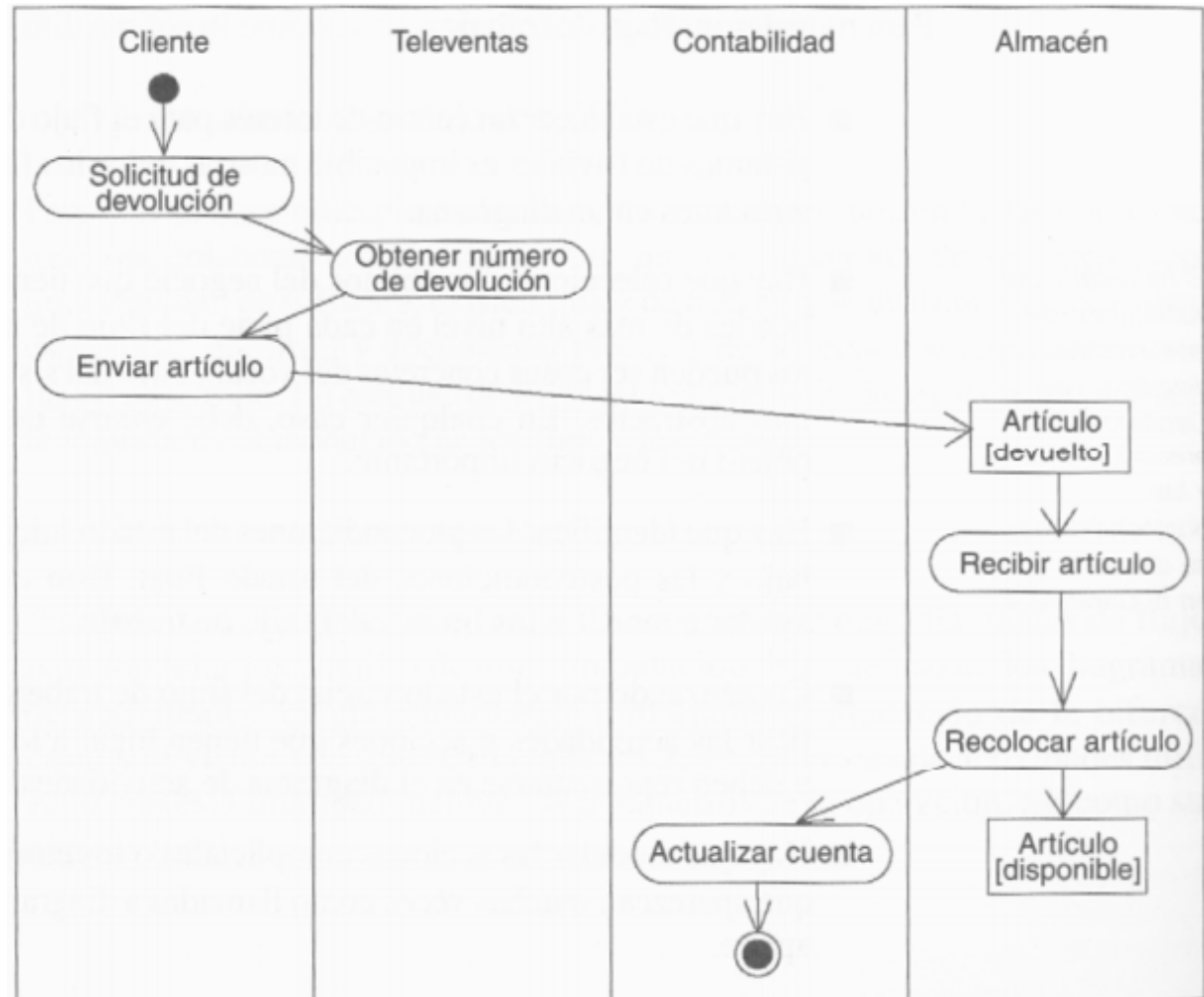
- Los aspectos dinámicos pueden involucrar el **comportamiento dirigido por eventos** de cualquier tipo de **objeto**
- Preferibles para modelar objetos reactivos

- **Diagrama de Actividades**

- Para modelar aspectos dinámicos de **una actividad** que puede **involucrar diversos clasificadores** en del sistema
- Preferibles para modelar un flujo de trabajo o una operación.

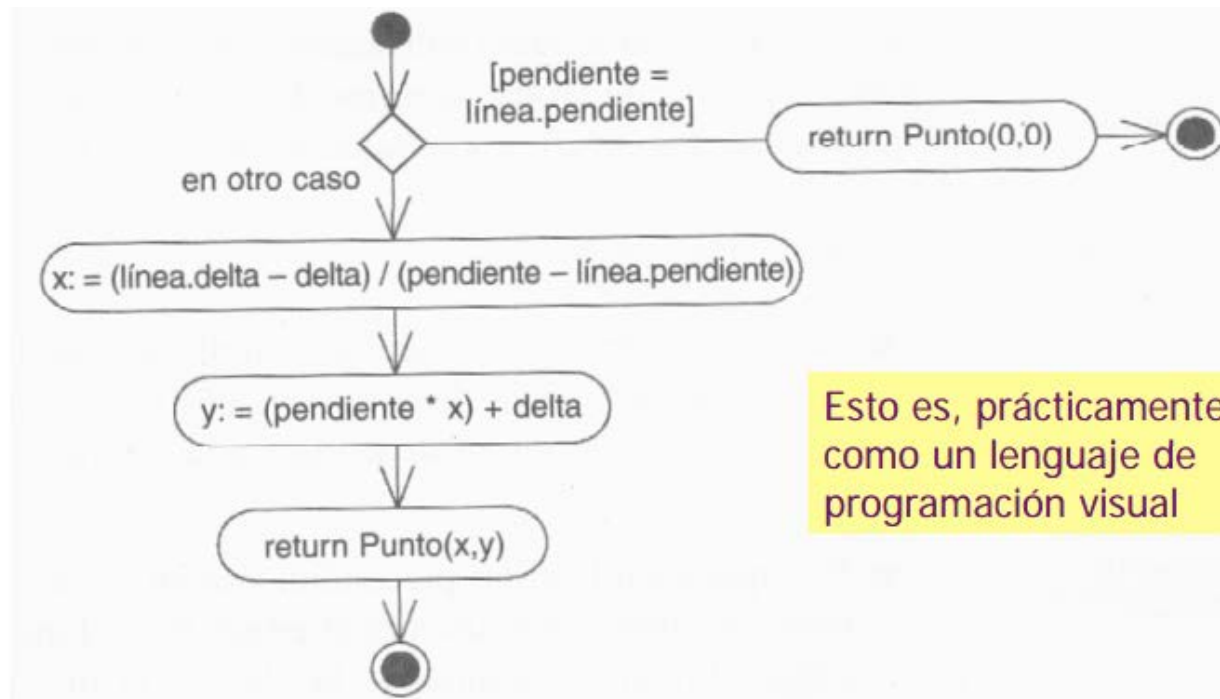
Ejemplo

- Diagrama de actividades para modelar un flujo de trabajo del sistema:
“Negocio de Venta”



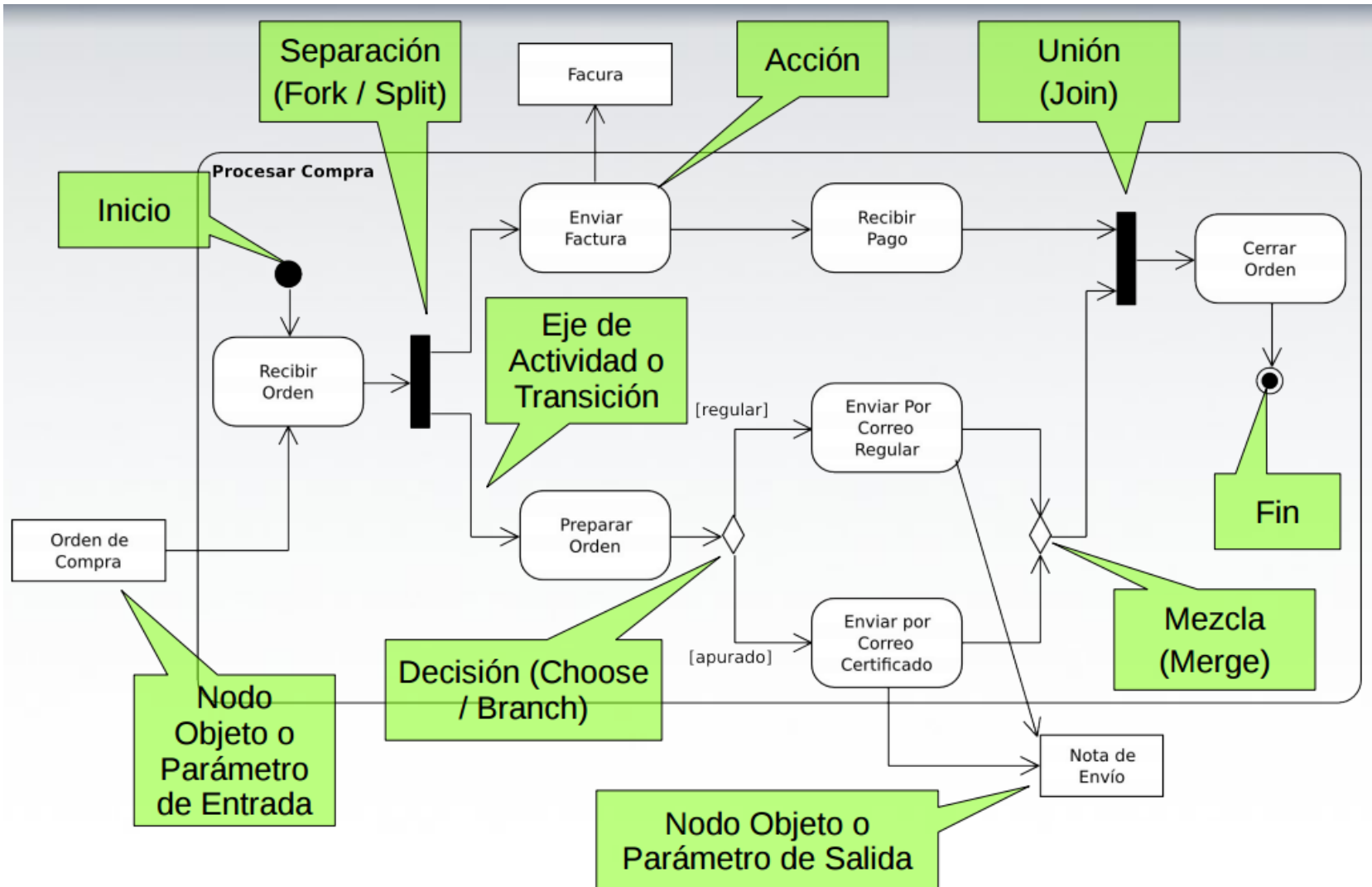
Ejemplo

- Diagrama de actividades para modelar una operación de una clase: “Intersección” de la clase “Línea”

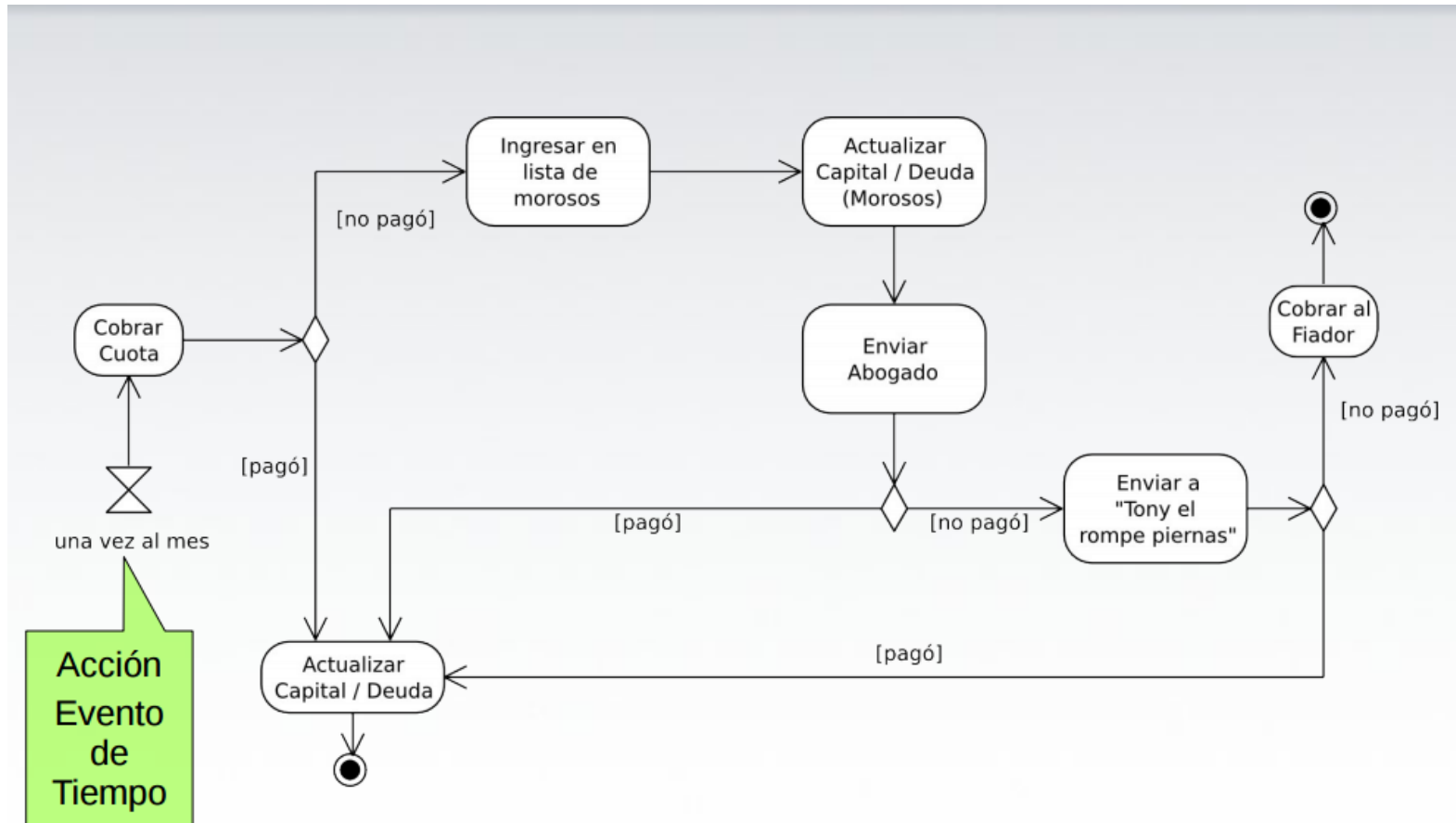


Esto es, prácticamente, usar UML como un lenguaje de programación visual

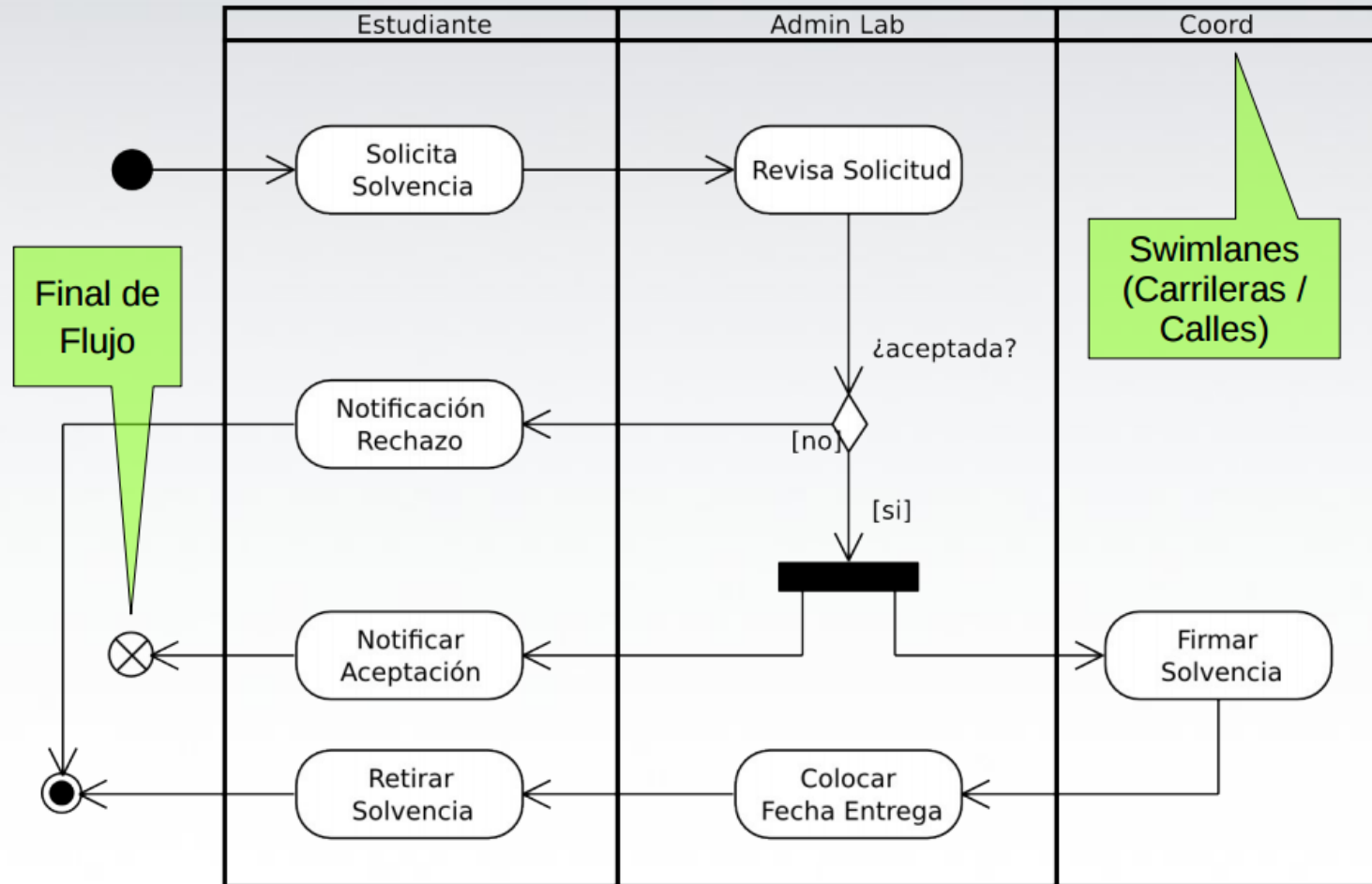
Ejemplo



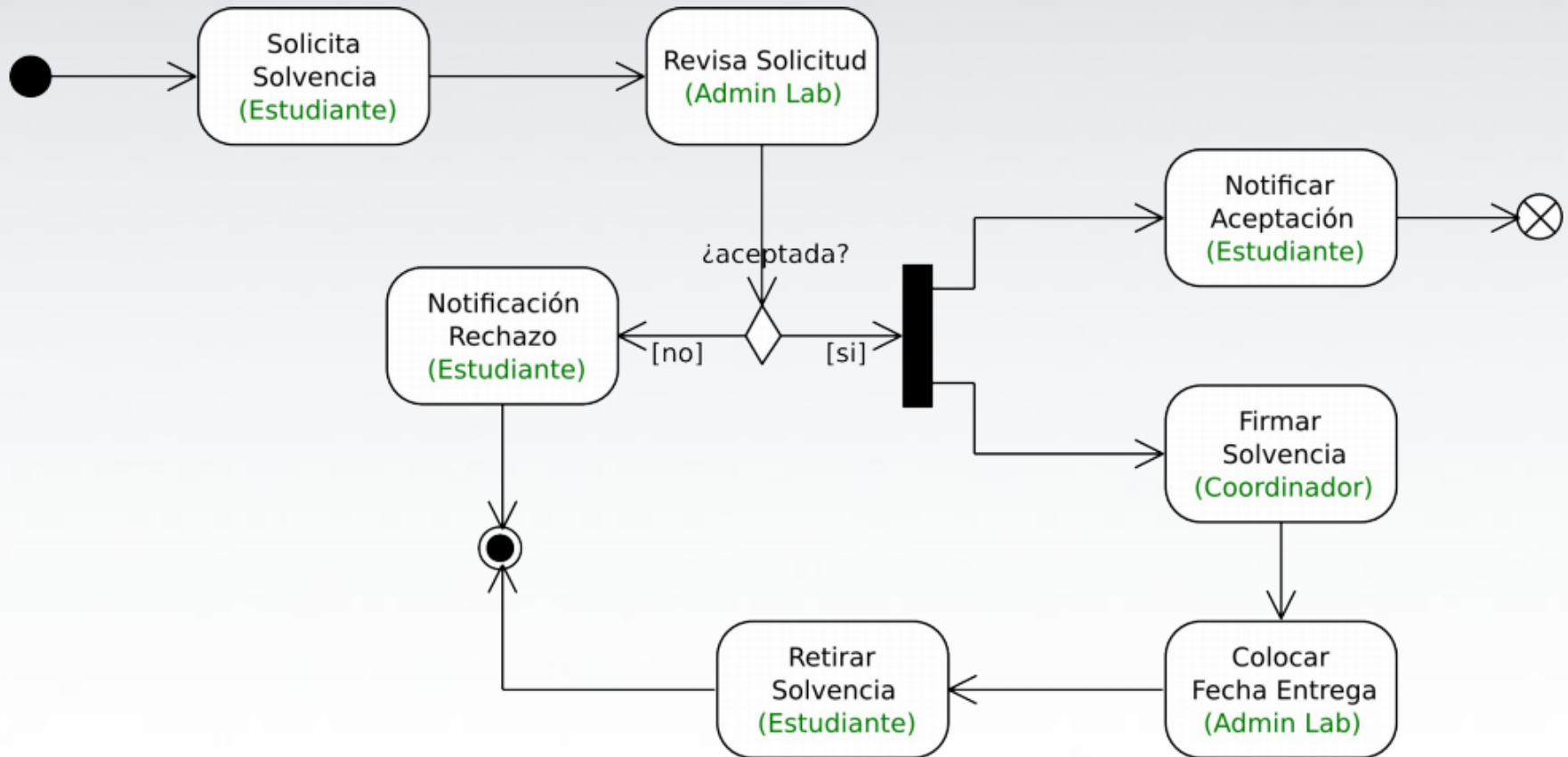
Ejemplo



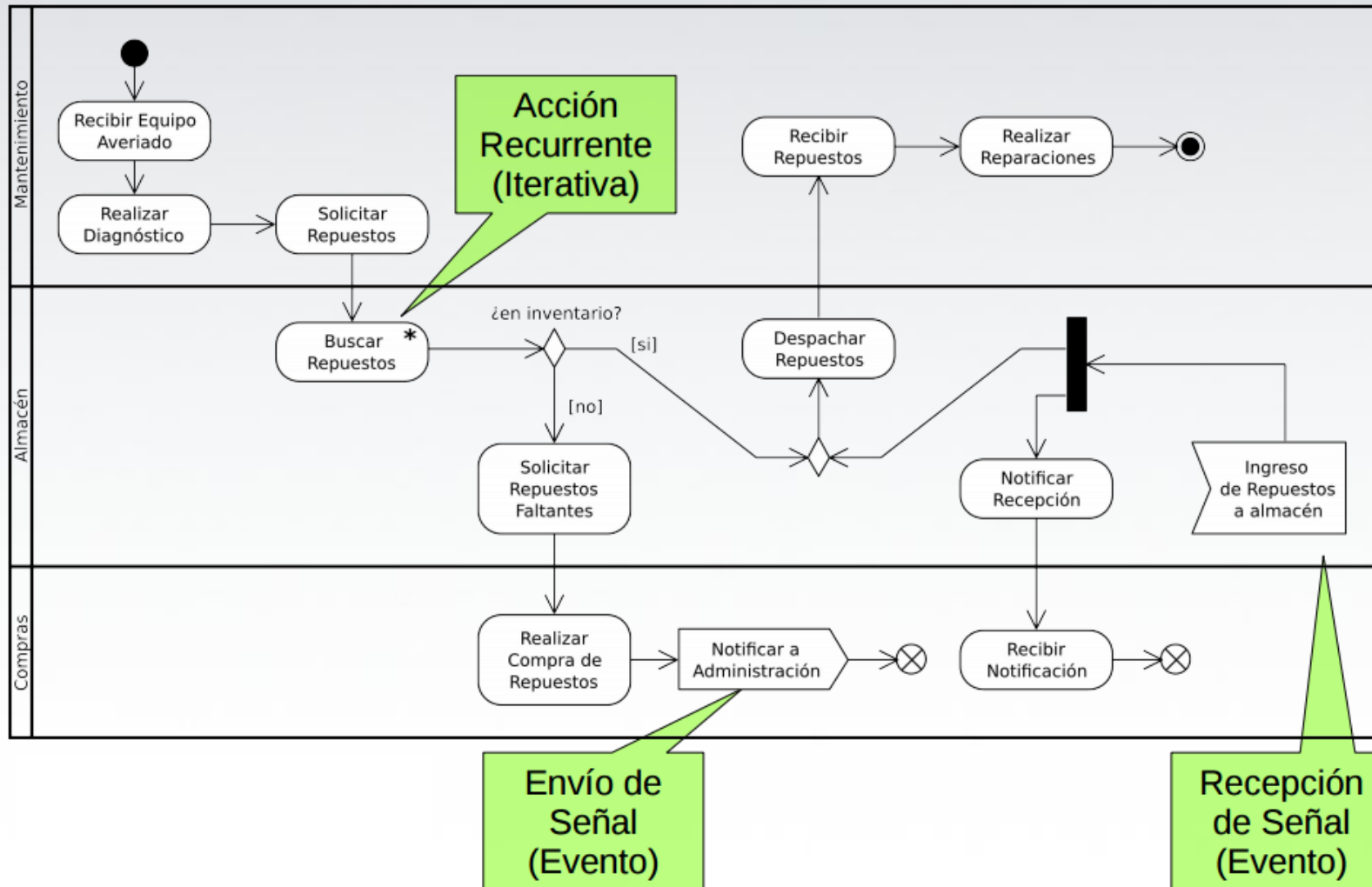
Ejemplo



Ejemplo



Ejemplo



Ejercicio

Nombre	UC-01. Añadir Nuevo enlace	
Precondición	No	
Secuencia principal	1	El visitante selecciona la opción de añadir un nuevo enlace.
	2	El sistema solicita la información del Nuevo enlace.
	3	El usuario introduce la información del enlace
	4	El sistema almacena el nuevo enlace
Alternativas / erróneas	2.1	Si sucede un error recuperando las categorías o no se encuentra ninguna categoría, el sistema muestra un mensaje de error y este caso de uso termina.
	3.1	En cualquier momento el usuario puede cancelar la operación y este caso de uso termina.
	3.2	Si los datos no son correctos, el sistema muestra un mensaje de error y se repite el paso 2.
	4.1	Si sucede un error almacenando el enlace, el sistema muestra un mensaje de error y este caso de uso termina.
Post condición	El Nuevo enlace se almacena en el sistema.	
Notas	No.	