

3. Diseño de Software

Ph.D Priscilla Jiménez P.

Conceptos de Diseño

El proceso de diseño
Atributos de la calidad
del software

Diseño de Software

- El diseño crea una representación del modelo del software.
- El modelo de los requerimientos se centra en describir datos que se necesitan, la función, y el comportamiento.
- El modelo de diseño proporciona detalles sobre la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema.

El objetivo del diseño es producir un modelo que tenga resistencia, funcionalidad y belleza.

Diseño del Software: Elementos

El diseño representa al software de varias maneras.

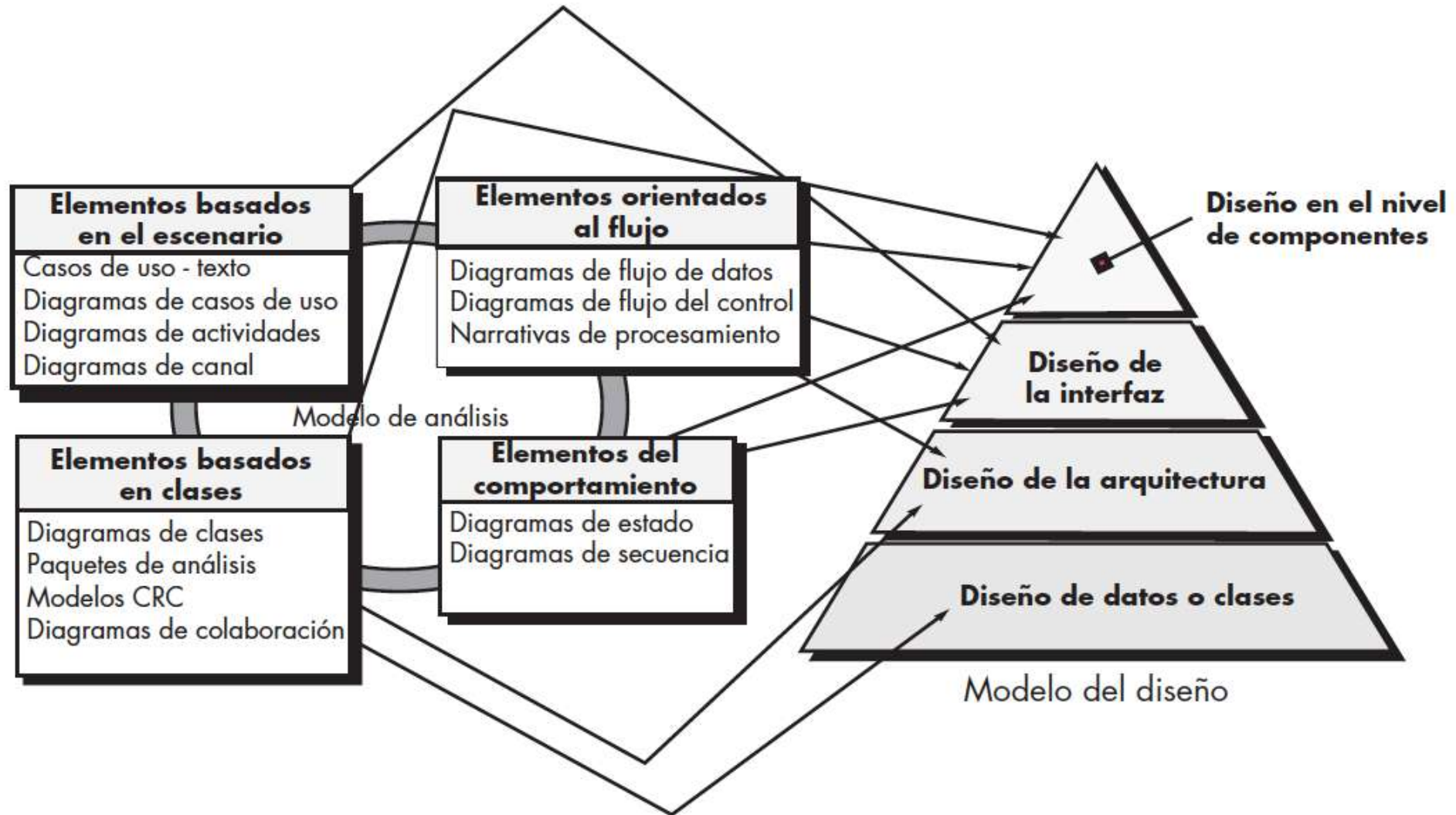
1. Diseño de Datos
2. Arquitectura del sistema o producto.
3. Interfaces que conectan al software con los usuarios finales, con otros sistemas y dispositivos.
4. Componentes del software que se utilizan para construir el sistema.

Cada perspectiva representa una acción de diseño distinta.

Diseño del Software

El modelo de software es evaluado por el equipo de software para determinar:

1. errores,
2. inconsistencias u omisiones,
3. si existen mejores alternativas,
4. si es posible implementar el modelo dentro de las restricciones, plazo y costo establecido.



Diseño del Software



La importancia del diseño de software se resume en la palabra “calidad”.



Sin diseño se corre el riesgo de obtener un sistema inestable, que no se pueda cambiar fácilmente o evaluar.

Características que funcionan como guías para evaluar un buen diseño

- Debe implementar todos los requerimientos explícitos contenidos en el modelo de requerimientos.
- Debe ser una guía legible y comprensible para quienes generan el código y para los que lo prueban y dan el apoyo posterior.
- Debe proporcionar el panorama completo del software, y abordar los dominios de los datos, las funciones y el comportamiento desde el punto de vista de la implementación.

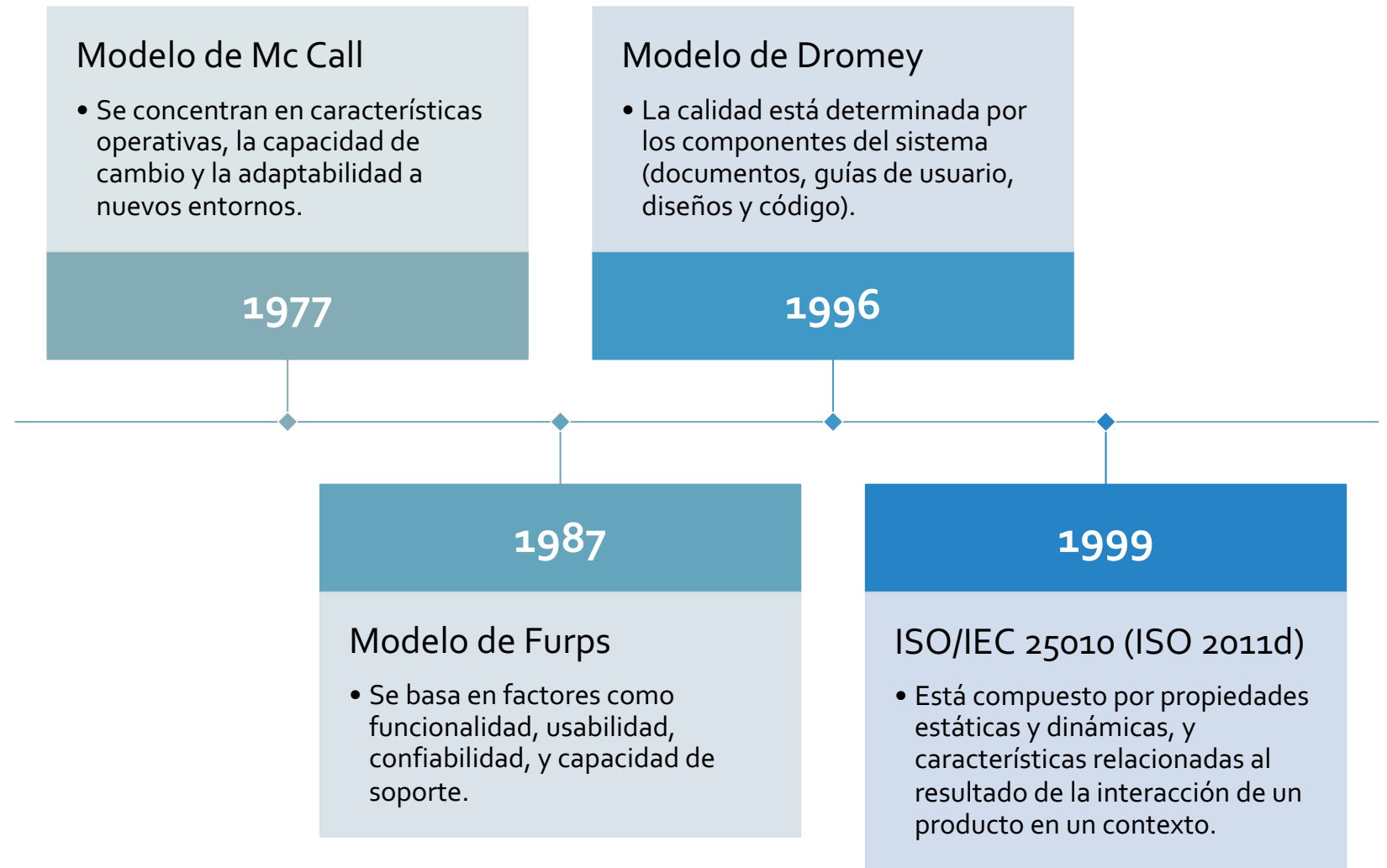
Lineamientos para el diseño

1. Debe ser modular.
2. Debe tener una arquitectura que se haya creado con el empleo de estilos o patrones arquitectónicos.
3. Debe contener distintas representaciones de datos, arquitectura, interfaces y componentes.
4. Debe conducir a estructuras de datos apropiadas para las clases que se van a implementar.

Lineamientos para el diseño

5. Los componentes deben tener características funcionales independientes.
6. Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los componentes y el ambiente externo.
7. Debe obtenerse con el empleo de un método repetible motivado por la información obtenida durante el análisis.
8. Deber representarse con una notación que comunique con eficacia su significado.

Estándares para la descripción de los factores de calidad



Calidad del Producto Software (ISO)

Adecuación Funcional

Compleitud, Corrección, y Pertinencia

Eficacia de desempeño

Comportamiento Temporal, Utilización de recursos y Capacidad.

Compatibilidad

Coexistencia e Interoperabilidad

Usabilidad

Percepción de la adecuación, Capacidad de aprendizaje, Operabilidad, Protección contra errores, Estética de la Interfaz, y Accesibilidad.

Fiabilidad

Madurez, Disponibilidad, Tolerancia a Fallos, Capacidad de Recuperación.

Seguridad

Confidencial, Integridad, Responsable

Mantenibilidad

Modularidad, Analizabilidad, Reusabilidad, Modificabilidad y Capacidad para ser probado

Portabilidad

Adaptabilidad, Capacidad para ser instalado, Capacidad para ser reemplazado.

Conceptos de diseño

Abstracción

Arquitectura

Patrones

División de
problemas

Modularidad

Ocultamiento
de
información

Independencia
funcional

Refinamiento

Aspectos

Rediseño

Clases de
diseño

Modelado del Sistema

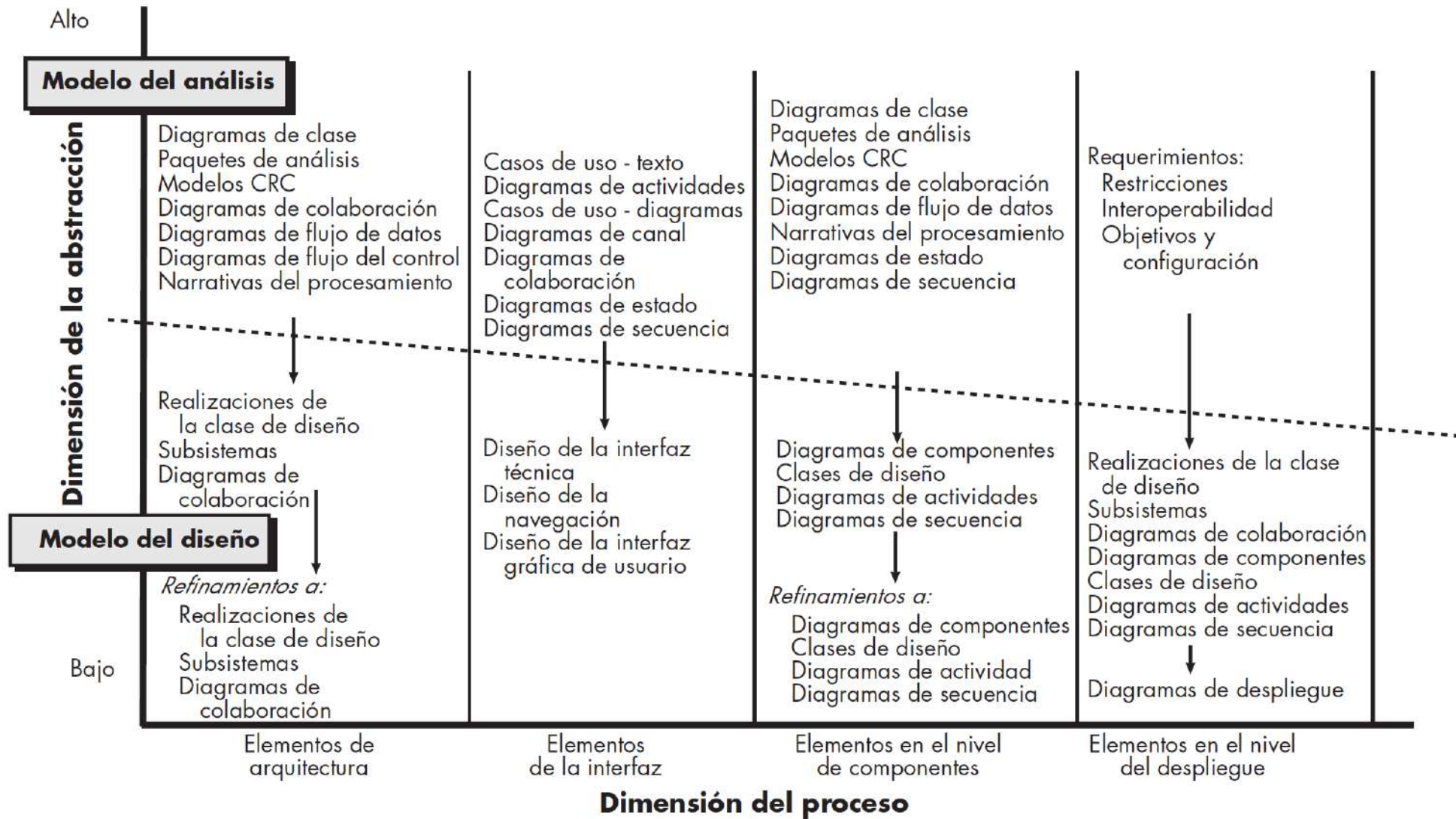
Dimensiones del
modelo de diseño

Modelos de contexto

Modelos de
interacción

Modelos estructurales

Modelos de
comportamiento



Dimensiones del modelo de diseño

Modelos del sistema

Son representaciones gráficas que describen los procesos del negocio, el problema a resolver y el sistema que tiene que ser desarrollado.

Los modelos representan el sistema desde diferentes perspectivas:

1. *Externa* -> se modela el contexto o entorno
2. *Interacción* -> se modela interacción entre un sistema y su entorno o componentes de un sistema.
3. *Comportamiento* -> se modela el comportamiento dinámico del sistema
4. *Estructural* -> se modela la organización del sistema o la estructura de datos.

Modelos del sistema - Usos

1. Como medio para facilitar discusión sobre un sistema existente o propuesto.
2. Como una forma de documentar un sistema existente.
3. Como una descripción detallada del sistema que sirve para general una implementación de sistema.

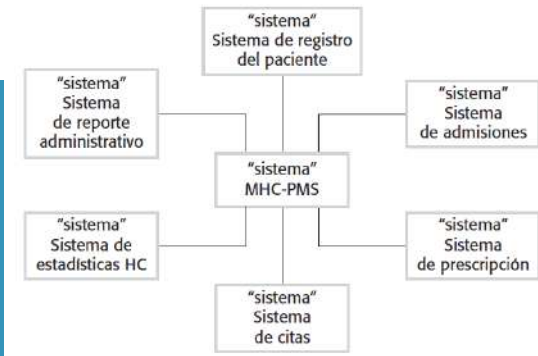
Tipos de diagramas frecuentemente usados.

- Diagramas de actividad, que muestran las actividades incluidas en un proceso.
- Diagramas de caso de uso, que exponen las interacciones entre un sistema y su entorno.
- Diagramas de secuencias, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema.

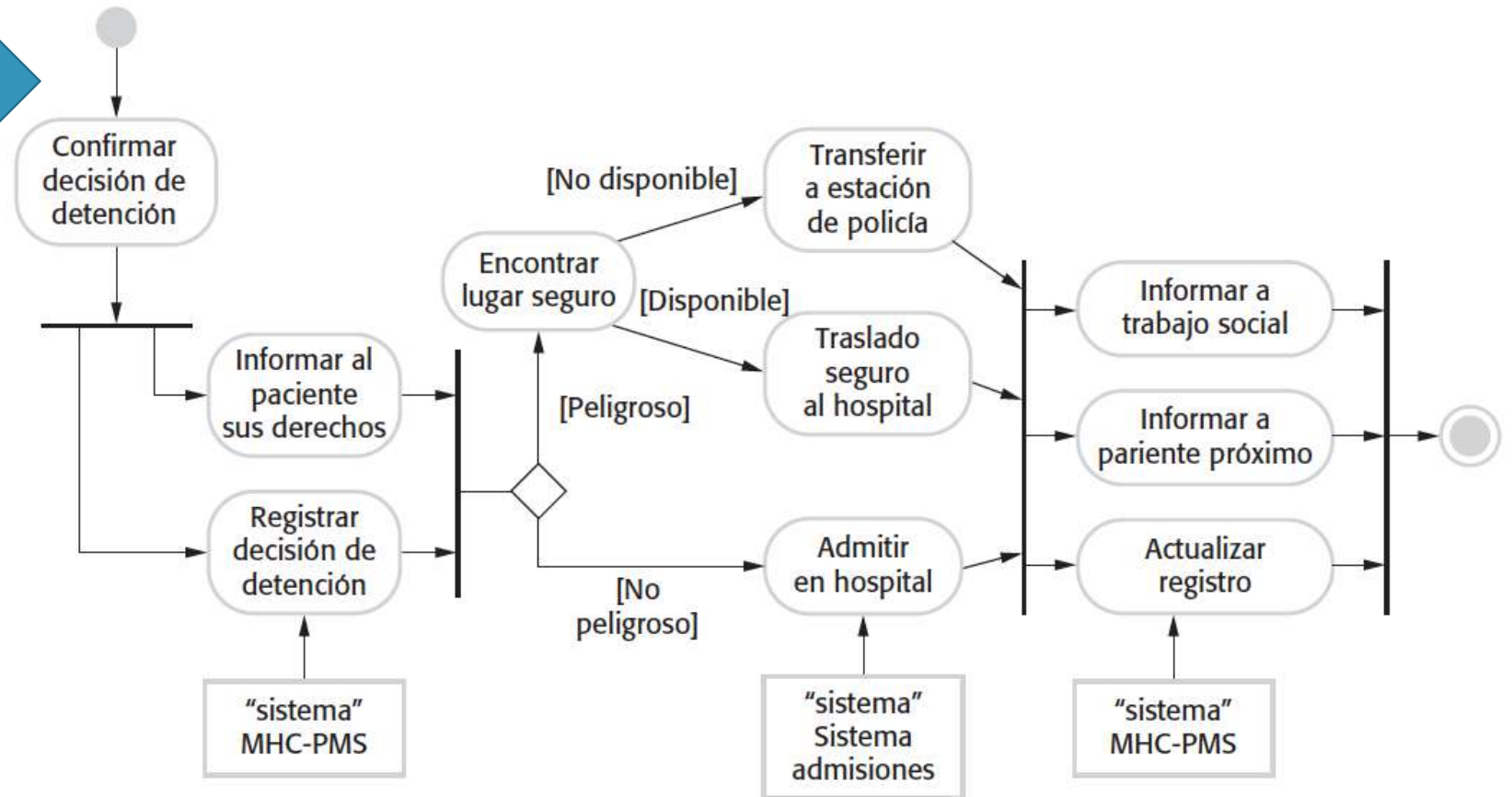
Tipos de diagramas frecuentemente usados.

- Diagramas de clase, que revelan las clases de objeto en el sistema y las asociaciones entre estas clases.
- Diagramas de estado, que explican cómo reacciona el sistema frente a eventos internos y externos.

Modelo de contexto



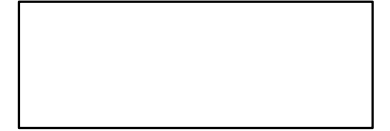
Uso de
diagrama de
actividad para
modelar
procesos.



- Acciones



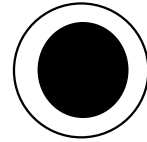
Objeto [Cambio de estado]



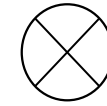
- Inicio



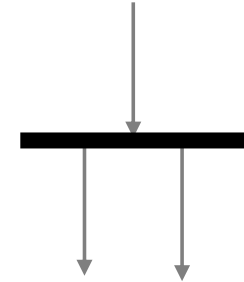
Fin



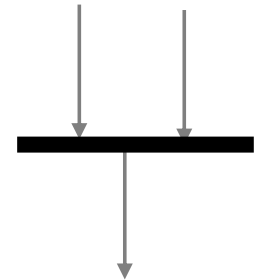
Fin de flujo



- Fork/División se usa para iniciar la concurrencia

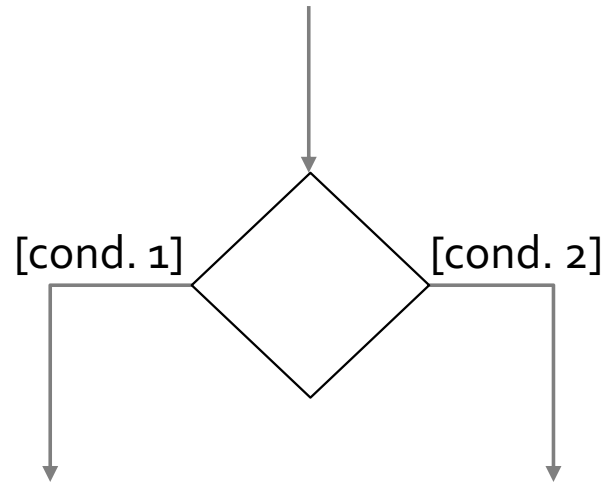


- Join/Unión es usado para regresar a un solo flujo

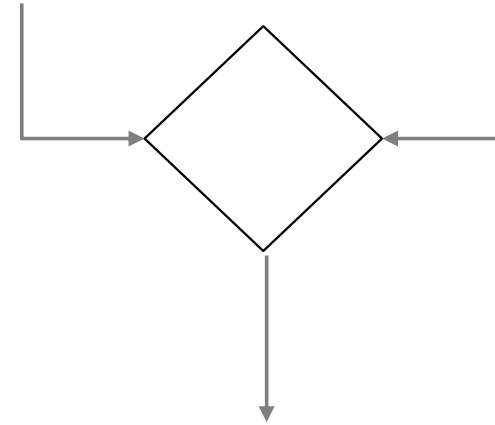


Diagramas de actividad

Bifurcaciones

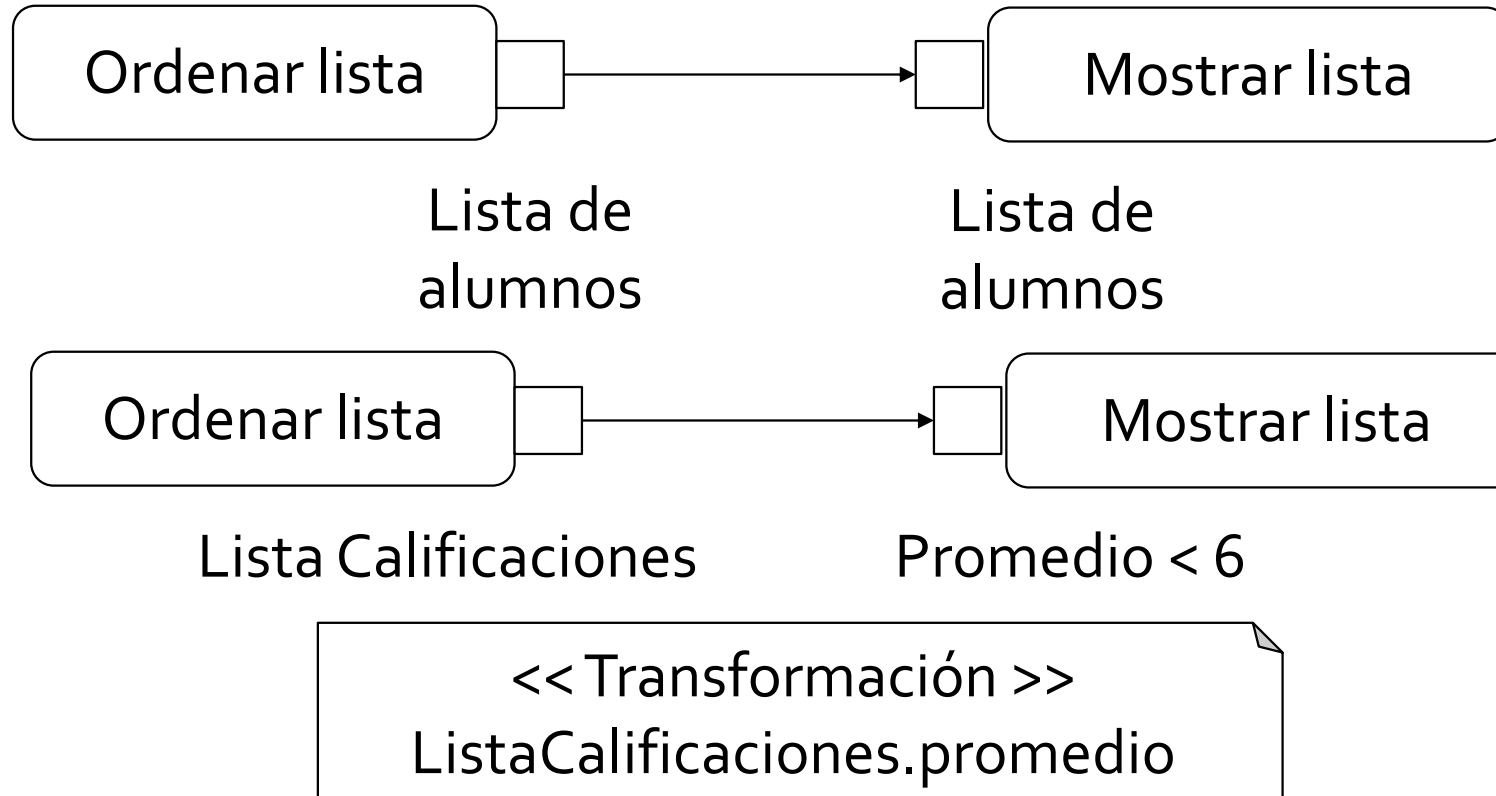


Fusión



Diagramas de actividad

Acciones que indican objetos de salida y objetos de entrada

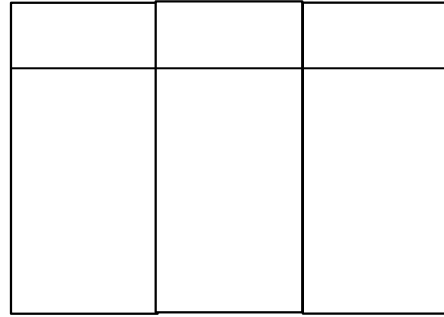


Diagramas de actividad

Subactividad



Particiones



Agrupar actividades que tengan algo en común (ej. departamento, ciudad, rol)

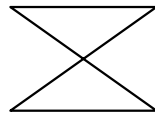
Enviar señal



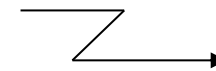
Acepta evento/señal



Señal de tiempo



Excepción

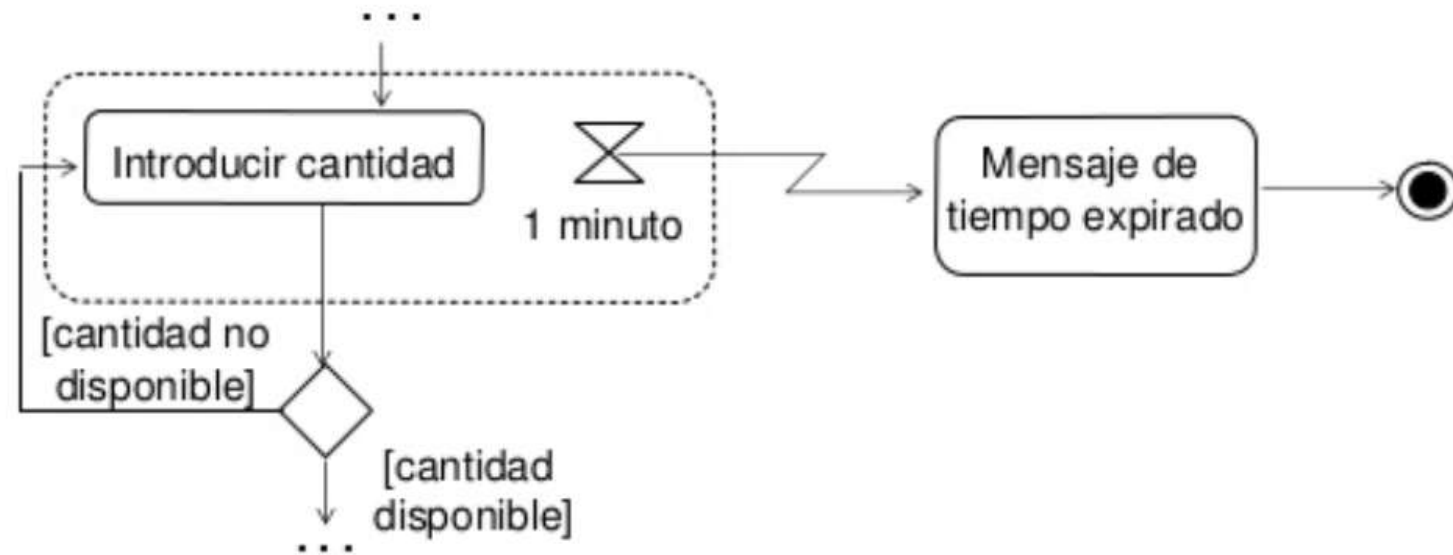


Diagramas de actividad

- Región que se puede interrumpir



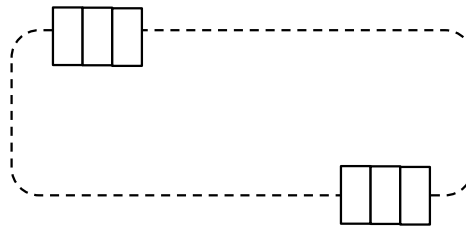
- Ej: ATM



Diagramas de actividad

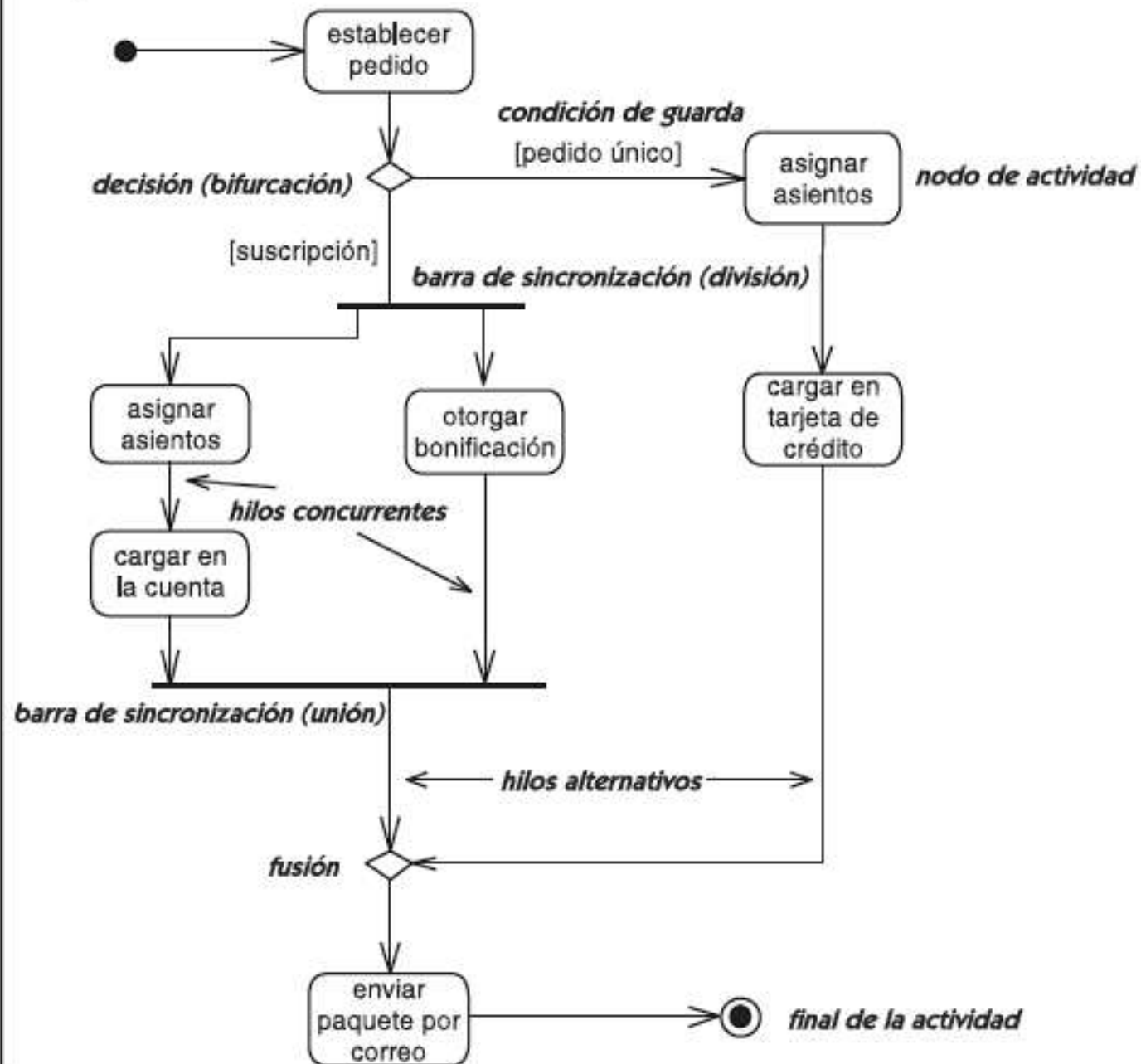
Flujo de una colección 

Región de expansión, es una región de actividad estructurada que se ejecuta muchas veces.



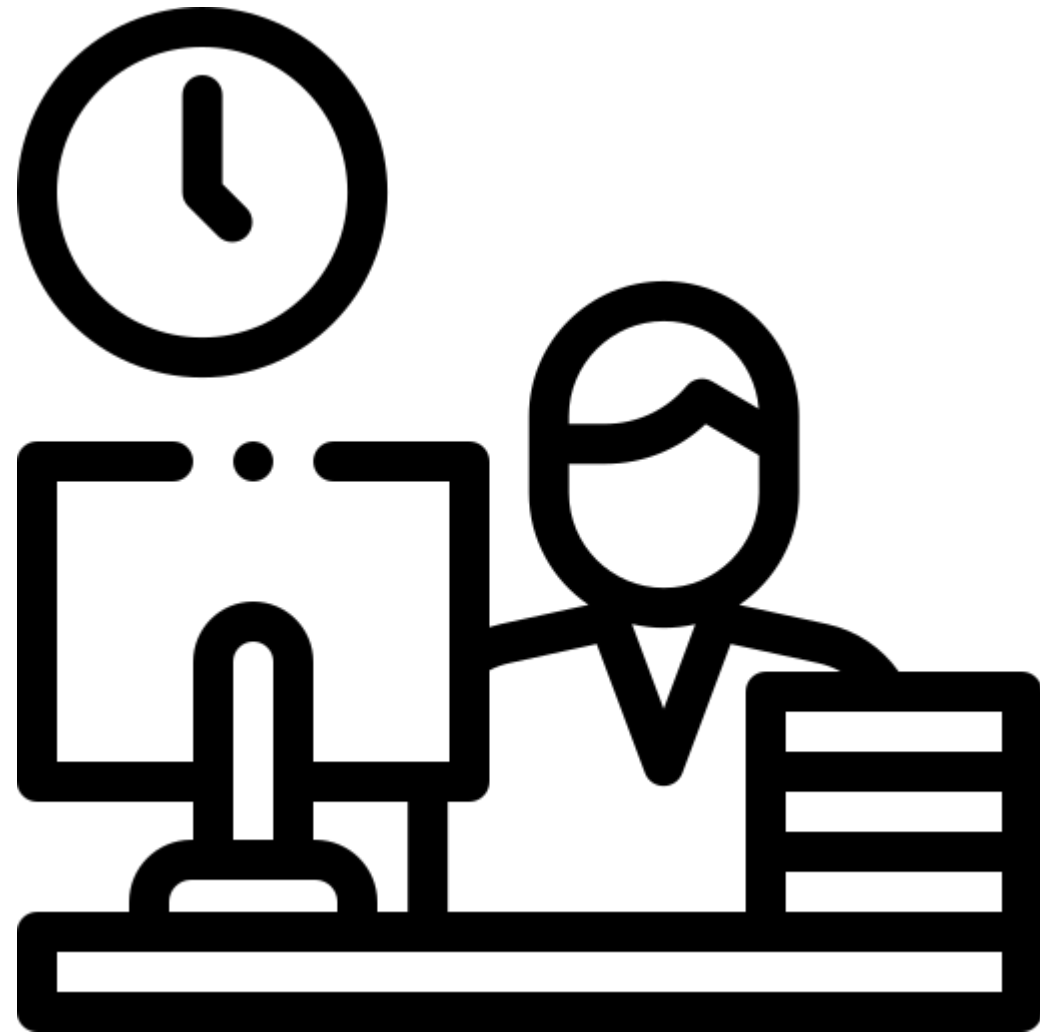
Diagramas de actividad

Taquilla::ProcesarPedido



Práctica 7

Diagrama de actividad



Modelos de interacción

Interacciones: entradas y salidas del usuario; interacciones entre el sistema a desarrollar y otros sistemas; o interacciones entre componentes del sistema.

1. Modelado de caso de uso

Modela interacciones entre un sistema y sus actores externos

2. Diagramas de secuencia

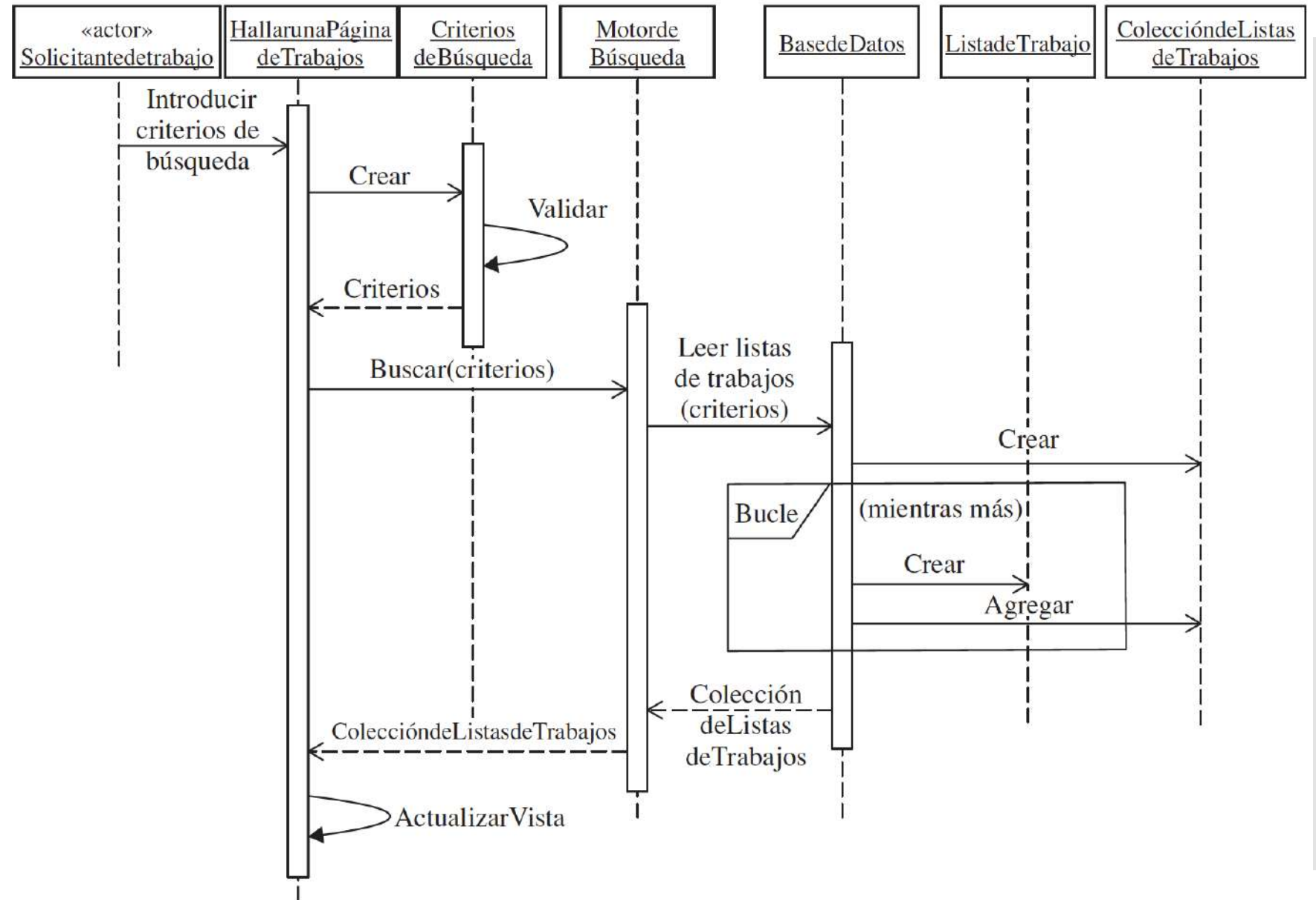
Modela interacciones entre componentes del sistema

Modelos de interacción: Diagramas de Secuencia

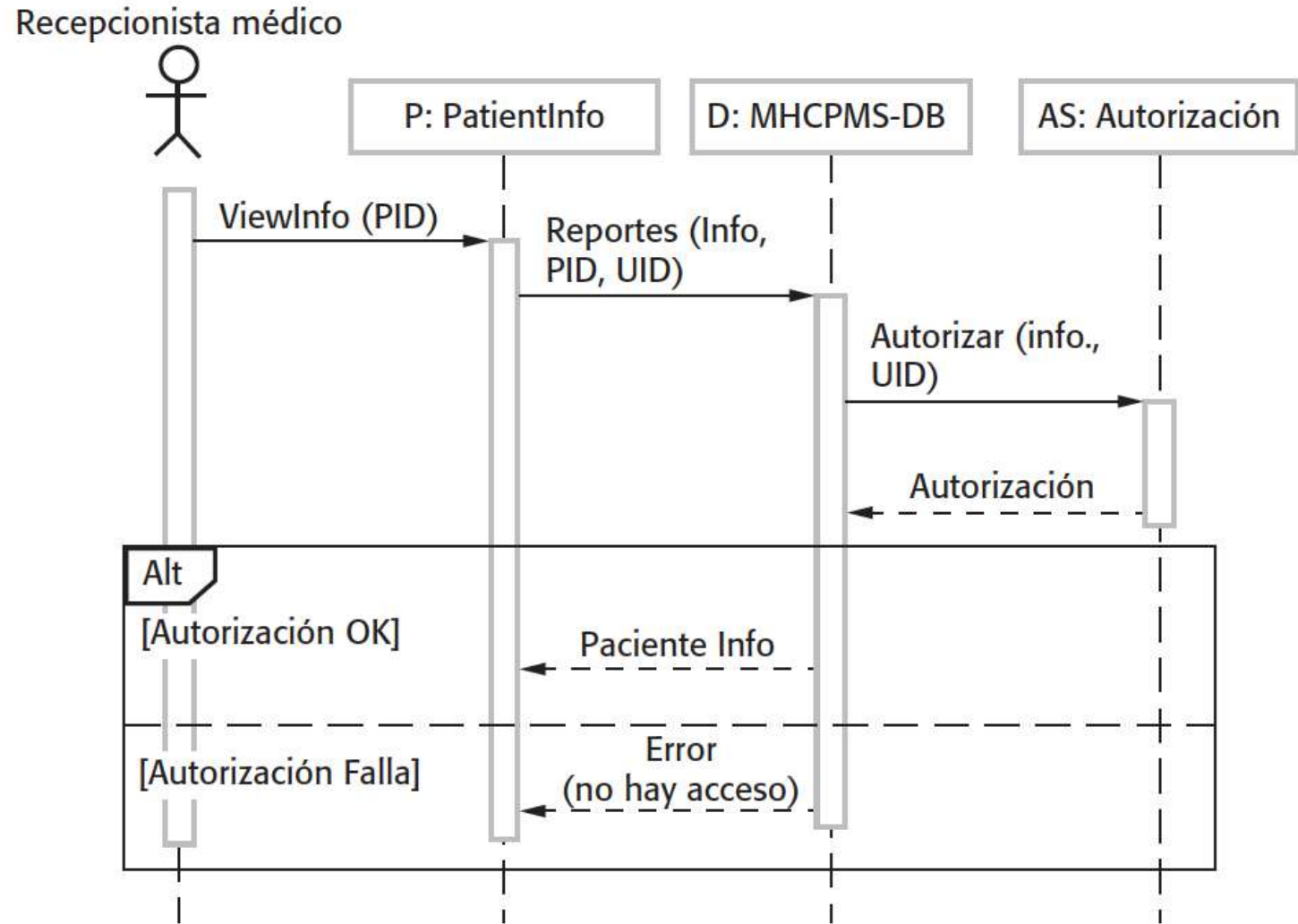
- Muestra la sucesión de interacciones que ocurre en una instancia de caso de uso.
- Describe el comportamiento dinámico del sistema.
- Indica la forma en que eventos provocan transiciones de un objeto a otro.
- El tiempo se mide en dirección vertical hacia abajo.

El orden es importante

Ejemplo: Diagrama de secuencia: Búsqueda de trabajos según criterio.



Ejemplo: Diagrama de secuencia: Ver información del paciente.



Modelos estructurales

Muestran la organización de un sistema en términos de los componentes y sus relaciones.

Diagramas de Clase - Proceso de construcción.

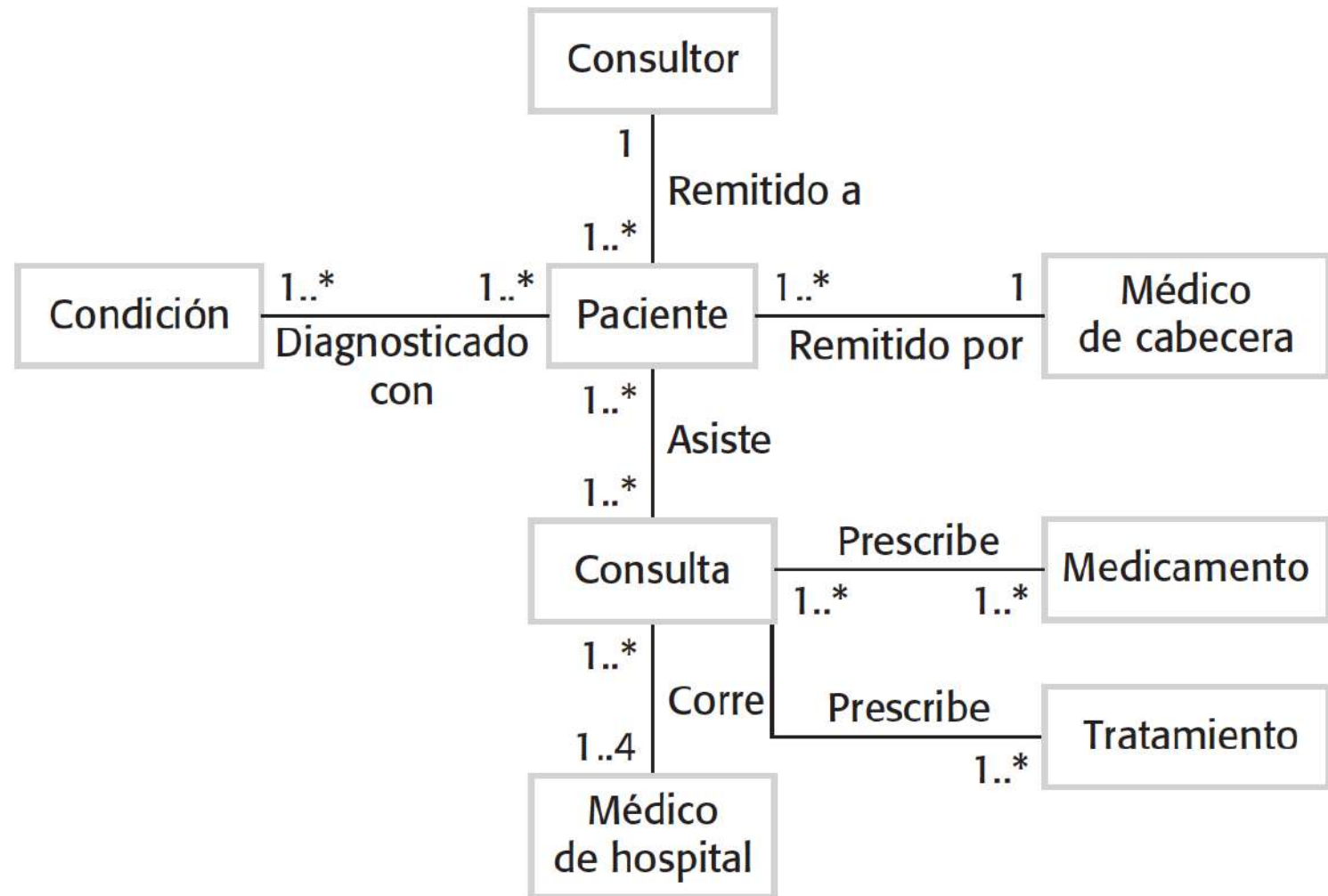
- Identificación de objetos y clases
- Identificación de atributos y funciones
- Identificación de las asociaciones y agregaciones
- Identificación de las relaciones de herencia
- Verificación y Refinamiento

Diagramas de Clase

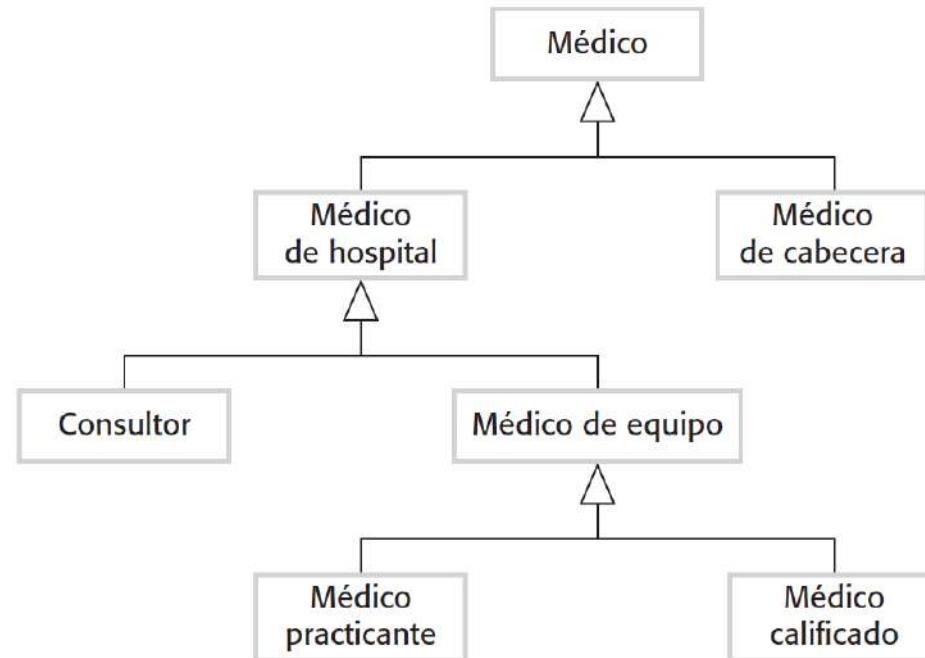
- Una clase es una definición general de un tipo de objeto del sistema.
- En las primeras etapas del proceso de ingeniería de software, los objetos representan algo en el mundo real.
- Conforme se desarrolla el sistema se necesitan añadir los objetos de implementación adicionales necesarios para una funcionalidad.

Se usan cuando se desarrolla un modelo de un sistema orientado a objetos.

Ejemplo de diagrama de clases

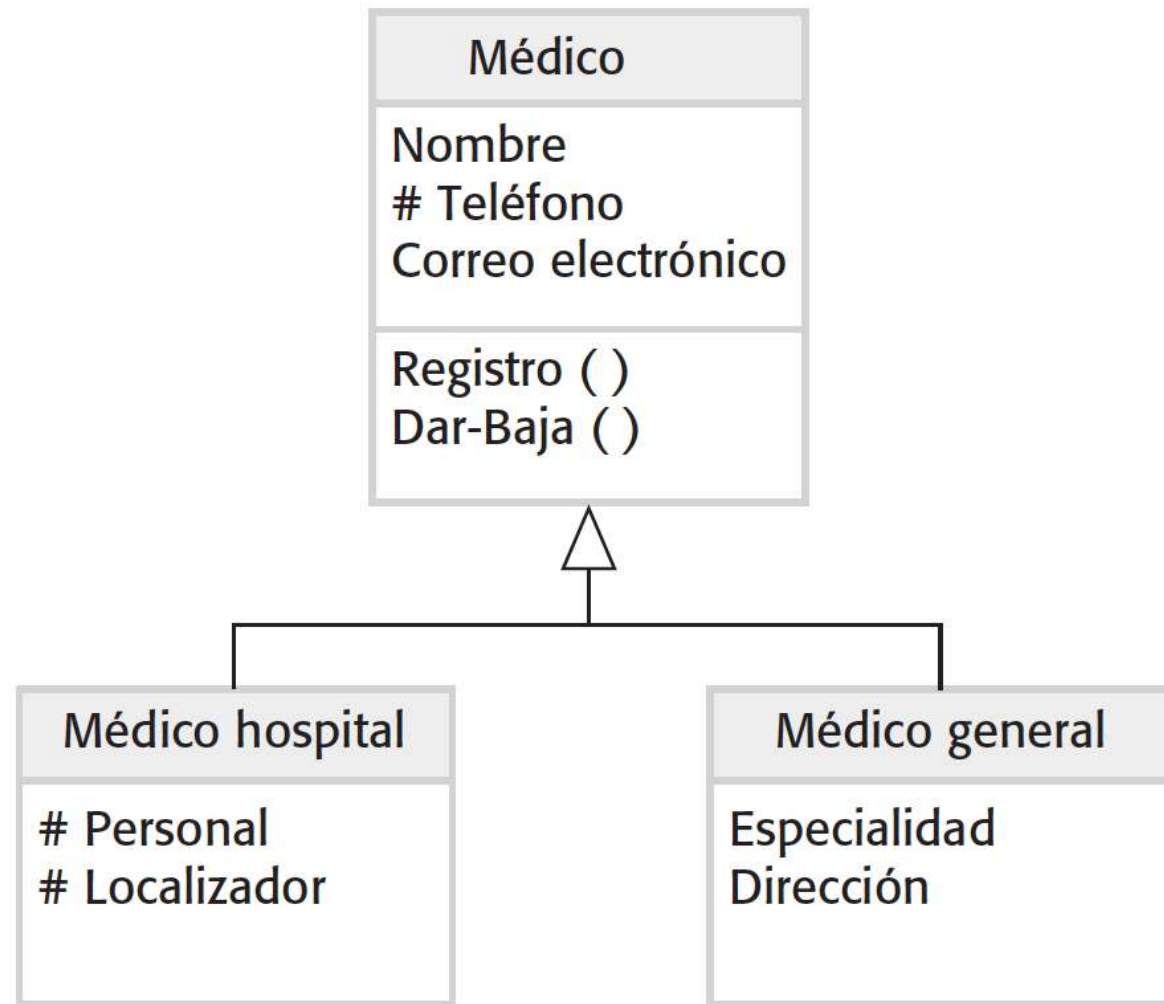


Ejemplo de Jerarquía de generalización y clase consulta



Consulta
Médicos Fecha Hora Clínica Razón Medicamento prescrito Tratamiento prescrito Notas de voz Transcripción ...
Nuevo () Prescribir () RegistroNotas () Transcribir () ...

Ejemplo de jerarquía con detalles agregados



Modelos de Comportamiento - Modelado dirigido por datos

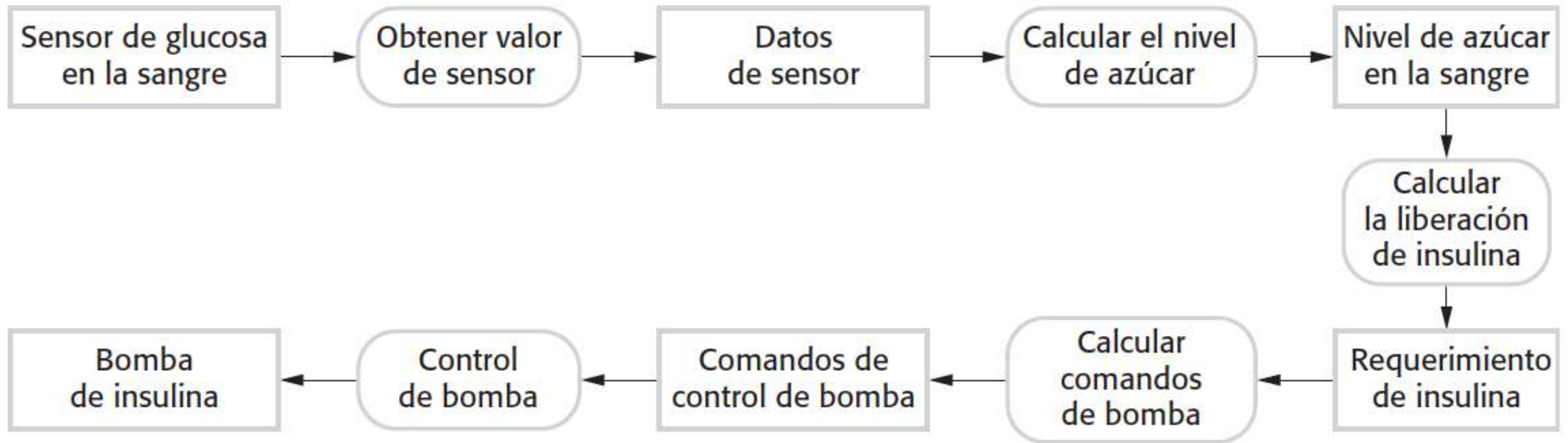
Muestran la secuencia de acciones involucradas en el procesamiento de datos de entrada, así como la generación de una salida asociada.



Diagramas de actividad (similares a los diagramas de flujo de datos).

Se pueden usar para modelar un caso de uso, una clase o un método complicado.

Una actividad produce alguna acción que produce algún cambio en el sistema.



Cadena de procesamiento de la operación de la bomba de insulina

Modelado dirigido por eventos

Muestra cómo responde un sistema a eventos externos e internos.

Se basa en la suposición de que un sistema tiene un número finito de estados y que los eventos (estímulos) pueden causar una transición de un estado a otro.

Se utilizan las máquinas de estado para modelar la aparición de un estímulo que puede disparar la transición de un estado a otro.

Modelado dirigido por eventos

Los
eventos
pueden
ser:

Síncronos: llamadas (invocación de operaciones)

Asíncronos: señales (excepciones), paso de tiempo, cambio de estado.

Según
donde
acontecen
pueden
ser:

Externos: si fluyen entre el sistema y sus actores (pulsación del ratón)

Internos: si fluyen entre objetos del sistema (una excepción)

Modelado dirigido por eventos

En UML se puede modelar 4 clases de eventos:

1. **Señal**, recepción de una comunicación asíncrona, explícita y con nombre entre objetos.

Input event >> User Input >> Mouse Button o
Keyboard character

Se modelan como clases estereotipadas con
<<señal>>

Para indicar que una operación envía una señal
<<enviar>>

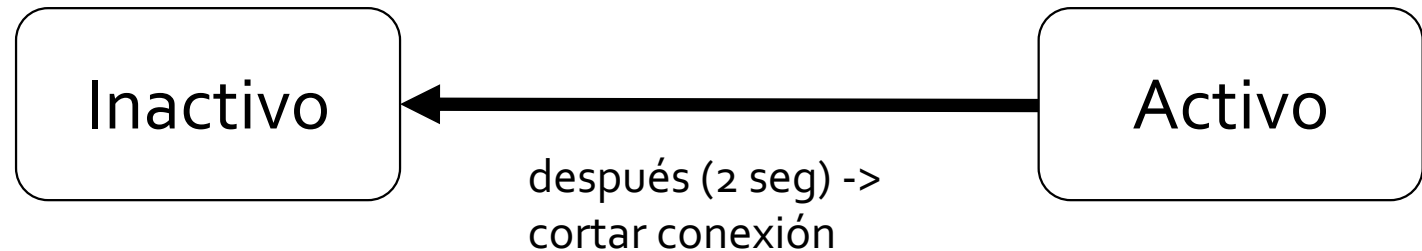
Modelado dirigido por eventos

2. *Llamada*, recepción por un objeto, de una petición explícita síncrona (cuando se invoca la operación se queda a la espera de recibir retorno).



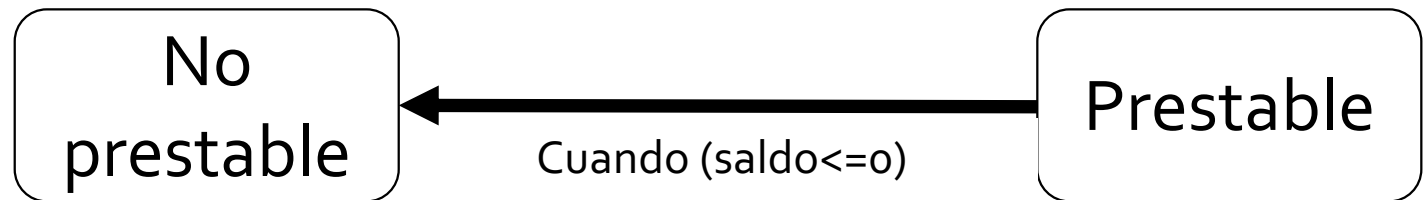
Modelado dirigido por eventos

3. *Tiempo*, llegada de un tiempo absoluto o transcurso de una cantidad relativa de tiempo. El evento representa un instante de tiempo mediante una expresión.



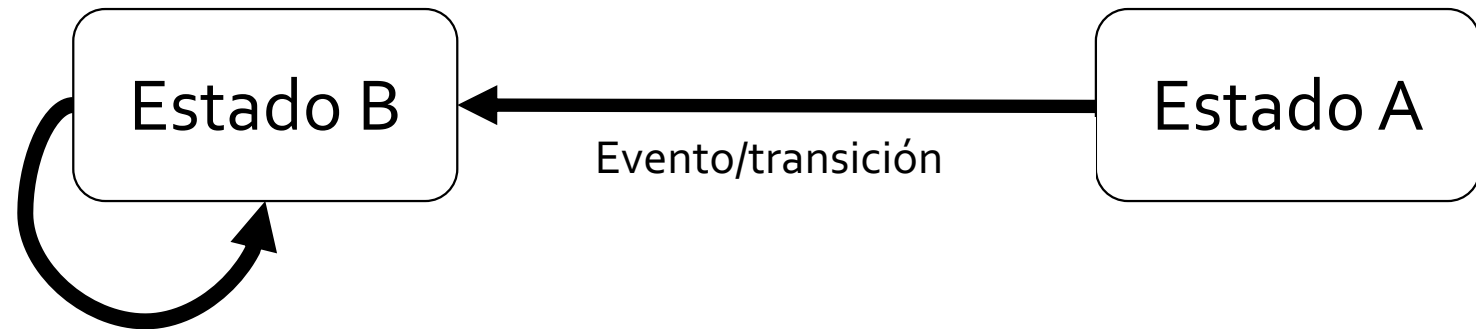
Modelado dirigido por eventos

4. **Cambio**, un cambio en el valor de una expresión booleana. Sólo se diagrama el cambio cuando el valor de la expresión cambia de falso a verdadero.



Máquina de estados

- 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 respuestas a eventos.



- Los nodos son los estados y los arcos las transiciones

Una máquina de estado tiene la forma de un grafo dirigido

Estado

- Representación de un objeto en los diferentes espacios de tiempo que le van sucediendo.
- Los diferentes estados que puede tener un objeto.
- Hay un estado inicial y un estado final.
- Cada evento representa algo que hace que nuestro objeto pueda cambiar.

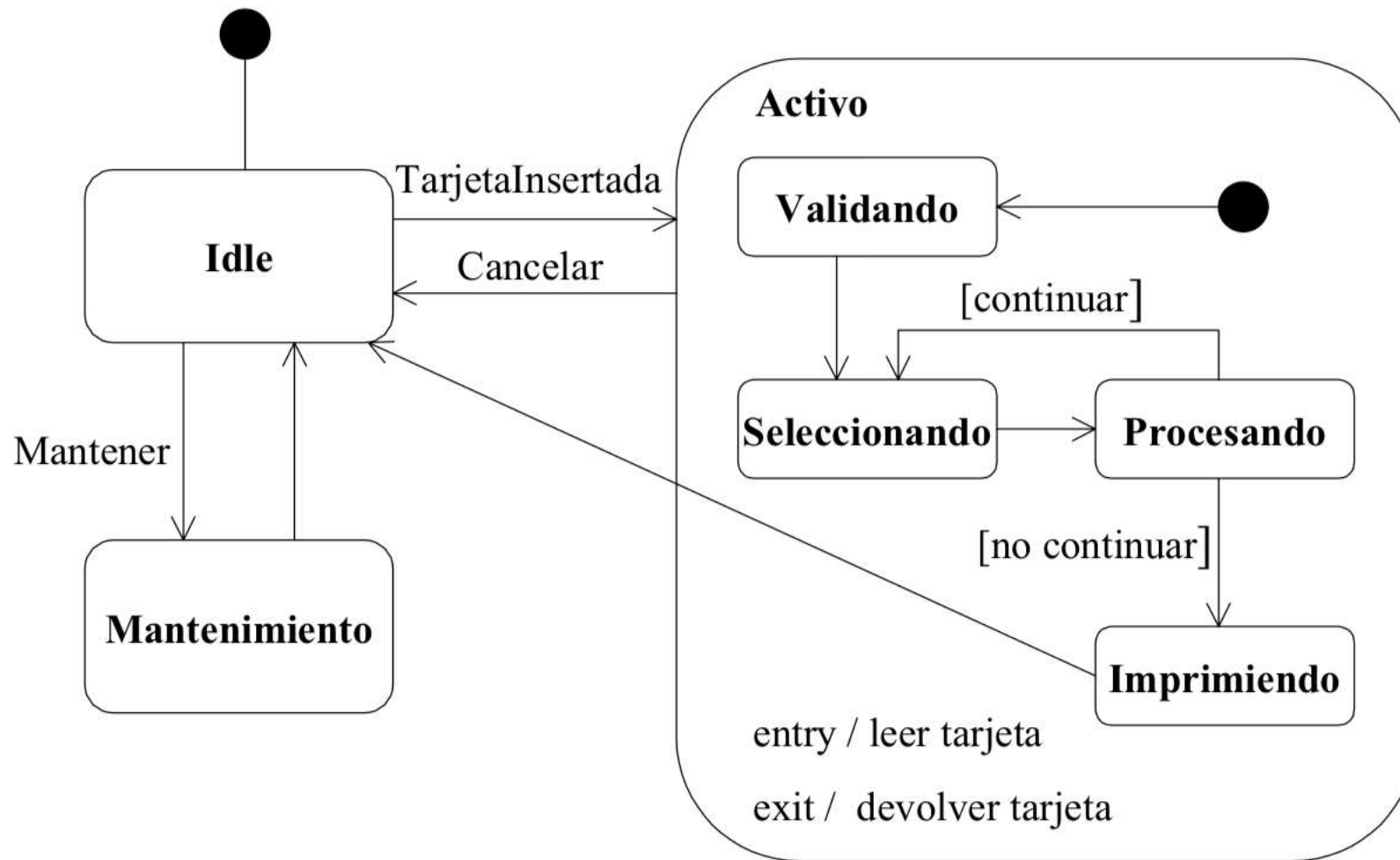
Partes de un estado

- Nombre (identificador)
- Acciones de E/S
- Transiciones internas (autotransiciones)
- Actividades/DO (actividades que se realizan mientras el estado esté activo)
- Eventos diferidos (eventos q se posponen para ser manejados por otro estado)

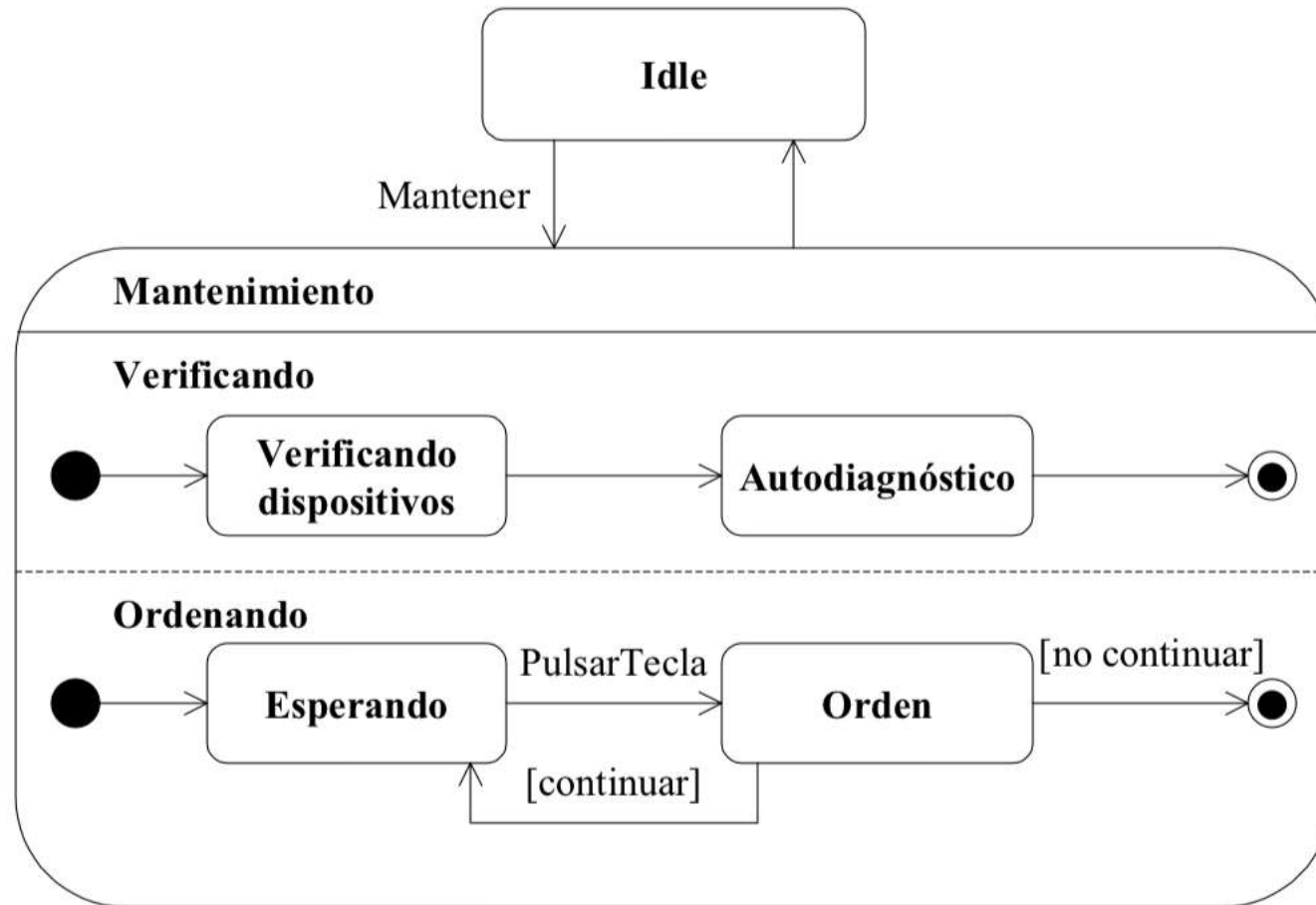
Rastreando

entry/ activarModo(enRastreo)
exit/ activarModo(noRastreo)

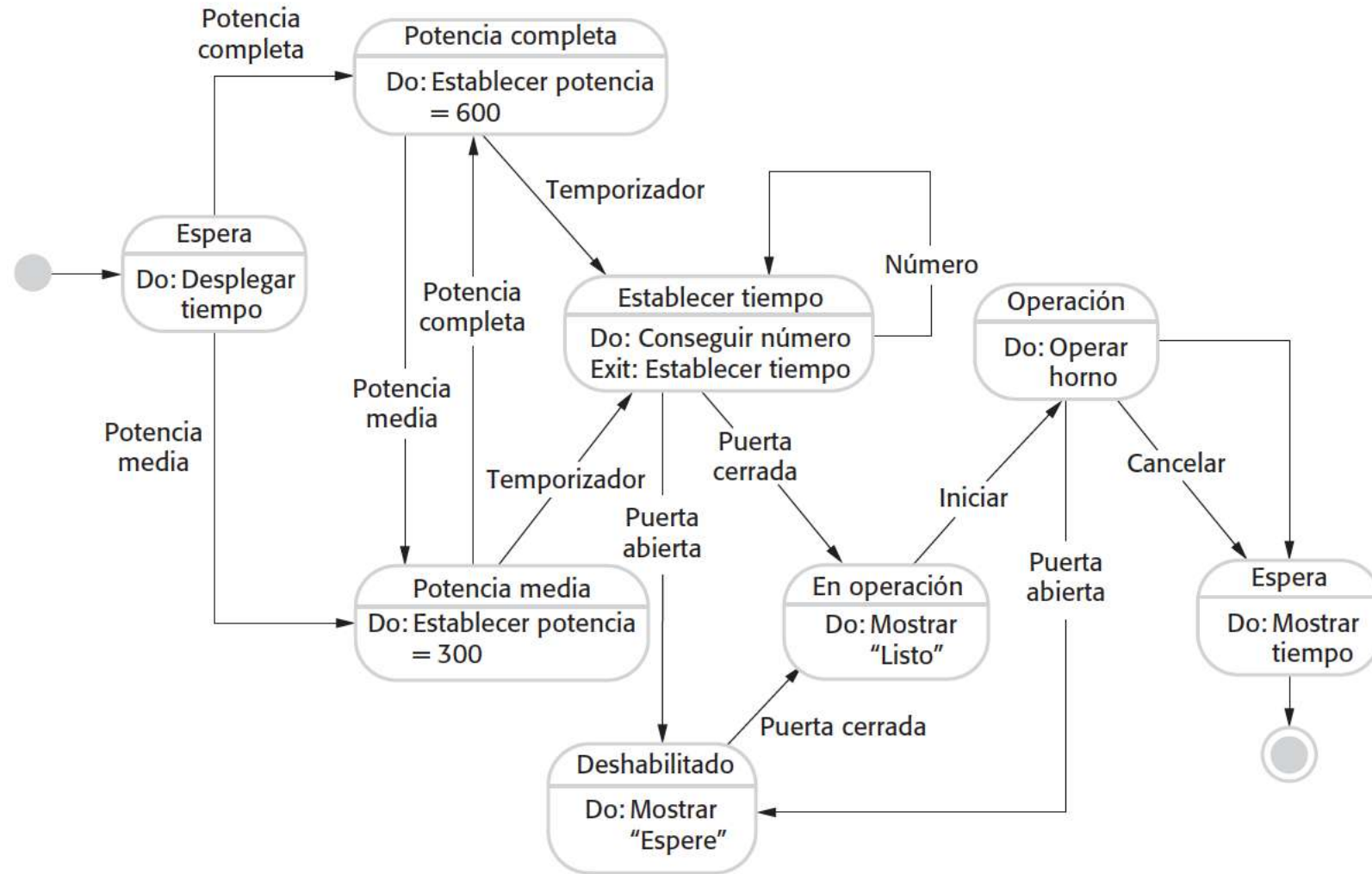
nuevoObjetivo/rastreador.Adquirir()
Do/seguirObjetivo
Aplazar/Autotest



Ejemplo - diagrama de estado de un cajero automático



Ejemplo - diagrama de estado de un cajero automático –
Estado mantenimiento



Ejemplo - diagrama de estado de un horno microondas

Estado	Descripción
Esperar	El horno espera la entrada. La pantalla indica el tiempo actual.
Potencia media	La potencia del horno se establece en 300 watts. La pantalla muestra "Potencia media".
Potencia completa	La potencia del horno se establece en 600 watts. La pantalla muestra "Potencia completa".
Establecer tiempo	El tiempo de cocción se establece al valor de entrada del usuario. La pantalla indica el tiempo de cocción seleccionado y se actualiza conforme se establece el tiempo.
Deshabilitado	La operación del horno se deshabilita por cuestiones de seguridad. La luz interior del horno está encendida. La pantalla indica "No está listo".
Habilitado	Se habilita la operación del horno. La luz interior del horno está apagada. La pantalla muestra "Listo para cocinar".
Operación	Horno en operación. La luz interior del horno está encendida. La pantalla muestra la cuenta descendente del temporizador. Al completar la cocción, suena el timbre durante cinco segundos. La luz del horno está encendida. La pantalla muestra "Cocción completa" mientras suena el timbre.

Estímulo	Descripción
Potencia media	El usuario oprime el botón de potencia media.
Potencia completa	El usuario oprime el botón de potencia completa.
Temporizador	El usuario oprime uno de los botones del temporizador.
Número	El usuario oprime una tecla numérica.
Puerta abierta	El interruptor de la puerta del horno no está cerrado.
Puerta cerrada	El interruptor de la puerta del horno está cerrado.
Iniciar	El usuario oprime el botón Iniciar.
Cancelar	El usuario oprime el botón Cancelar.

Descripción tabular – Estados y estímulos

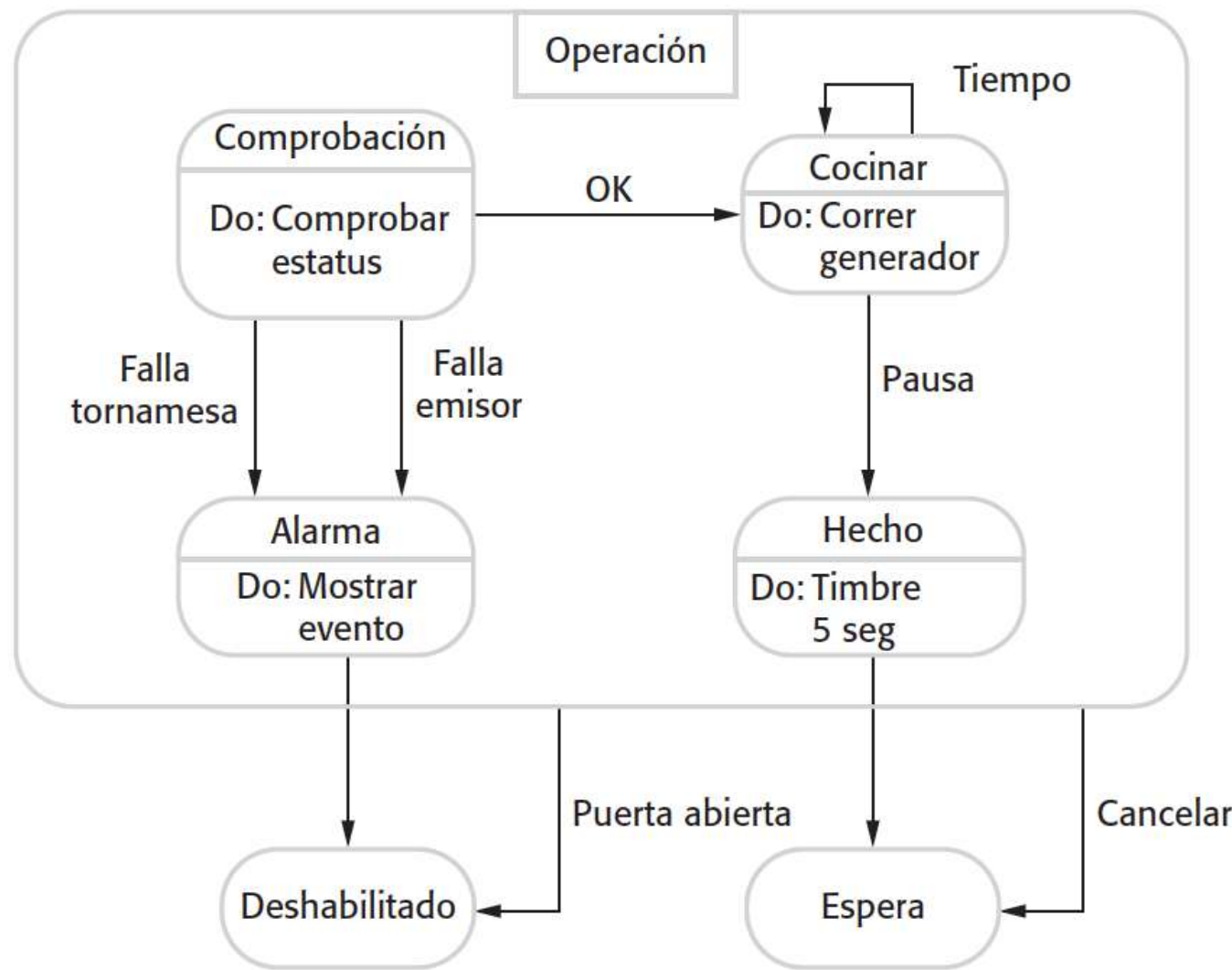


Diagrama de estado – Estado Operación de un horno microondas

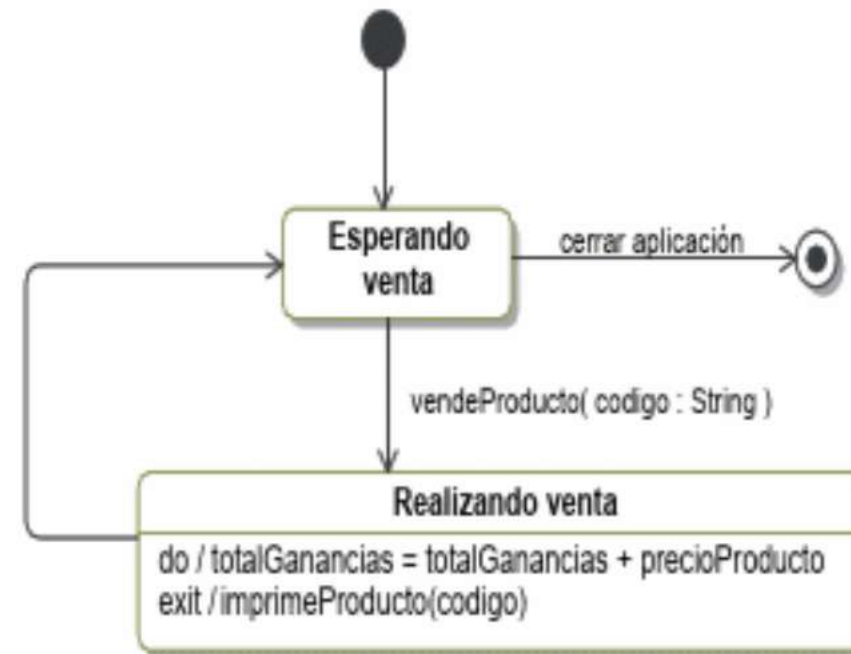
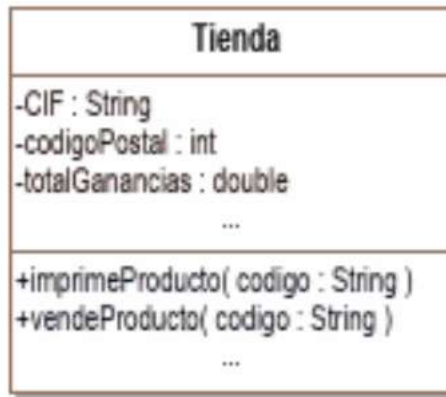
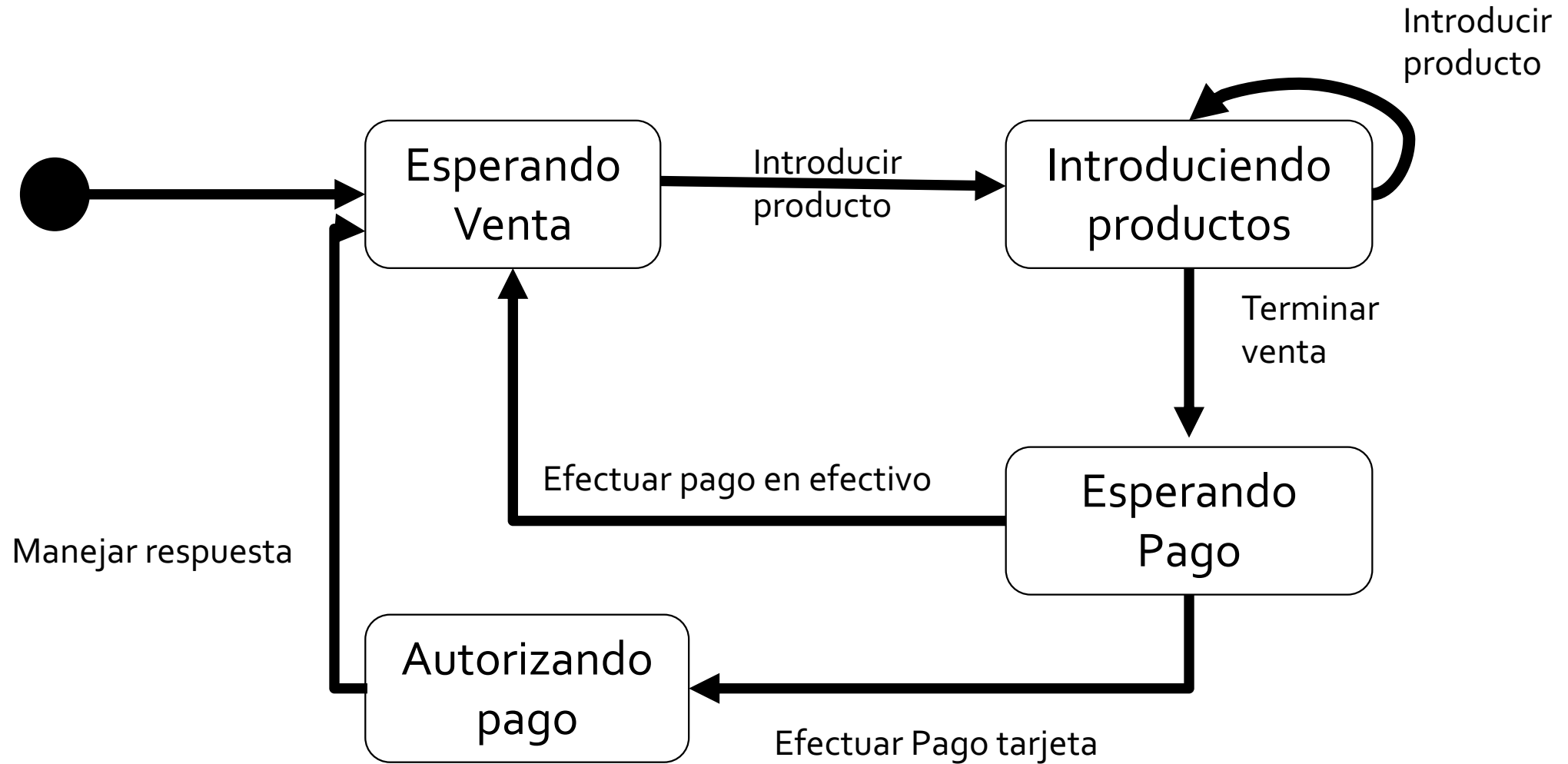


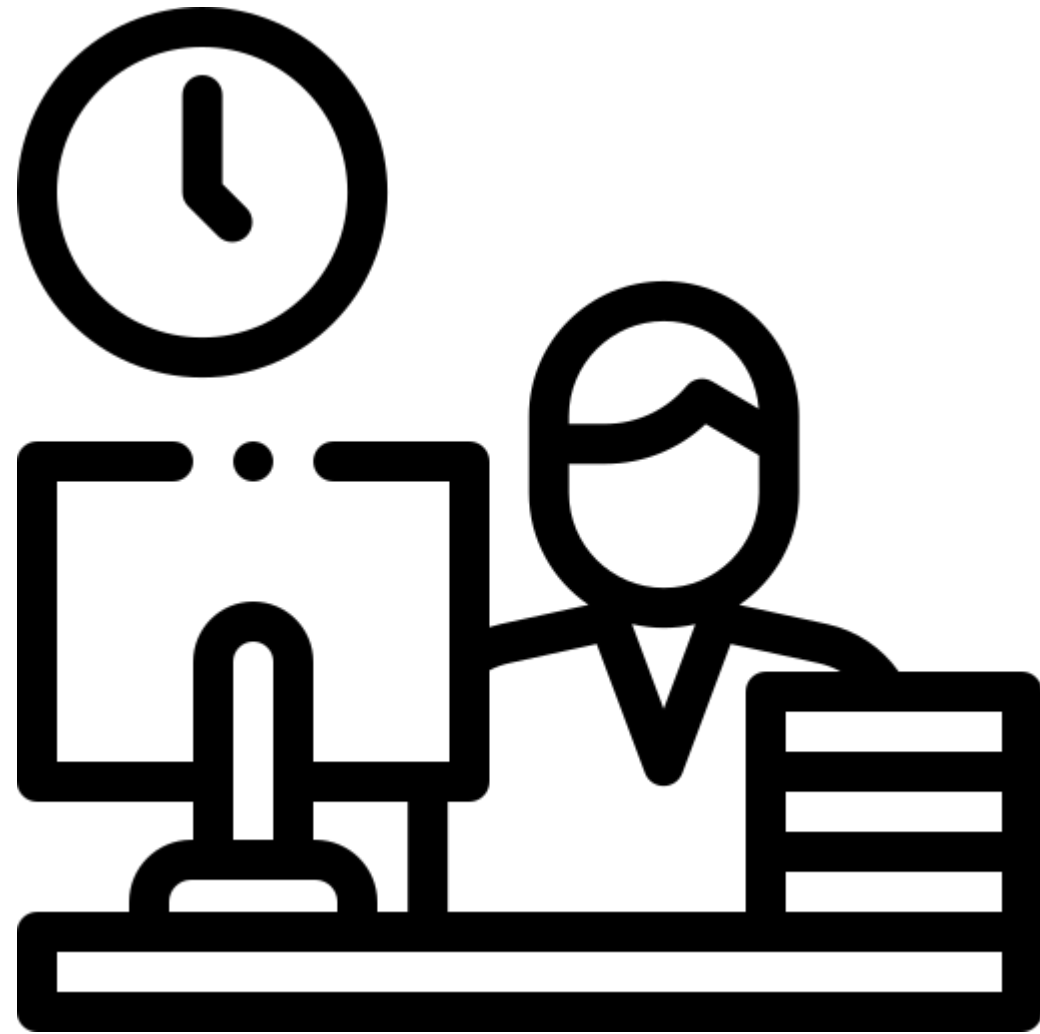
Diagrama de estado con eventos de llamada



Mostrar una tabla que defina el significado del estado y del estímulo.

Práctica 8

Diagrama de estados





Bibliografía y Recursos útiles

- Sommerville, I. (2011). Ingeniería de Software. 9^{na} Edición. (Cap. 5).
- Pressman, R. (2010). Ingeniería del Software. Un Enfoque Práctico. 7^{ma} Edición. (Cap. 8).
- Jiménez, de Parga, Carlos. *UML Aplicaciones en Java y C++*, RA-MA Editorial, 2014. *ProQuest Ebook Central*. (Cap 10,11)
- https://www.youtube.com/watch?v=Rk3cPADj__M
- <https://www.youtube.com/watch?v=j77wzk4hbMg>
- <https://www.youtube.com/watch?v=j77wzk4hbMg>
- <https://www.youtube.com/watch?v=y1pltD9z5Rs&t=475s>