

2º


Ingeniería del Software I

TEMA 4

Arquitecturas lógicas y físicas



TEMARIO

- TEMA 1 Principios básicos. Sentido y finalidad de la ingeniería del software
- TEMA 2 Procesos, modelos de proceso y ciclos de vida.
- TEMA 3.1 Ingeniería de requerimientos
- TEMA 3.2 Introducción desarrollo ágil e software
- TEMA 4 Arquitecturas lógicas y físicas 
- TEMA 5.1 Diseño arquitectónico
- TEMA 5.2 Diseño detallado e implementación
- TEMA 6 Pruebas de Software
- TEMA 7 Evolución

T4. Arquitecturas lógicas y físicas

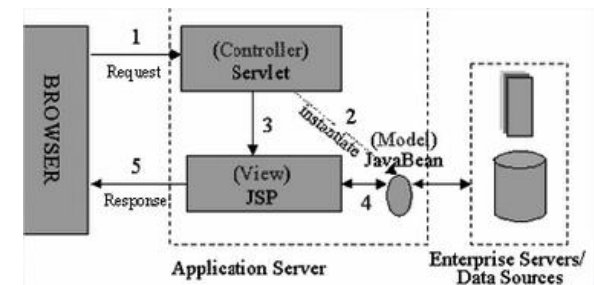
Arquitectura lógica VS Arquitectura física

Arquitectura Física:

- La **arquitectura física** expresa cuáles son los **componentes físicos** (cliente, servidor, servidor web, BD, etc.) que participan en nuestra solución, así como la **relación** entre ellos.
- La **especificación de la arquitectura física** normalmente consta de **uno o más diagramas**, y la **explicación** de los mismos (*actores y relaciones entre ellos*).
 - En la explicación de los diagramas se debe especificar el **nombre y la función** de cada actor, y el **tipo de relación** que existe entre ellos (si existe alguna). También se puede incluir ejemplo aclaratorios de puntos que puedan resultar abstractos.

Ejemplos de arquitecturas y sus componentes:

- Servlets: (Jserv) CGI (PHP) - módulos java que nos sirven para extender las capacidades de los servidores web.
- Usuario: HTML, XML, Java.
- Request (Web): HTTP, XML, Java, RMI (método de invocación remoto)
- Web Server: (Apache), DBM (métodos de acceso a Base de Datos), (SQL, XML)

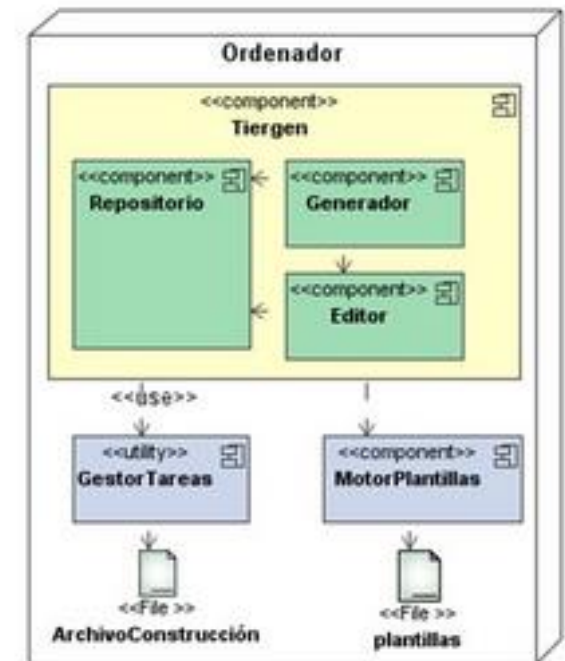


T4. Arquitecturas lógicas y físicas

Arquitectura lógica VS Arquitectura física

Arquitectura Lógica

- La arquitectura lógica expresa cuáles son los **componentes lógicos** (subsistemas, o macro-funciones) que participan en nuestra solución, y la relación entre ellos.
- La especificación de esta arquitectura es **similar a la arquitectura física**.
 - Se **especifican los actores y las relaciones** entre ellos, aunque los actores ahora son **subsistemas** de nuestra solución o **macro-funciones** de la misma.
- En los diagramas que expresan tanto la arquitectura lógica como la física, se puede utilizar casi **cualquier simbología que clarifique el escenario** (DFD, diagramas de clases, bloques, casos de uso, dibujo informal, etc.), a menos que existan restricciones al respecto.



- **Modelado de sistemas:** proceso para desarrollar **modelos abstractos** de un sistema, donde cada modelo presenta una **visión o perspectiva** diferente.
 - También es un **medio para representar el sistema** usando algún tipo de notación gráfica.
 - Generalmente utilizando notaciones **UML**.
- Es posible desarrollar modelos tanto del **sistema existente** como del **sistema a diseñar**.
 1. Modelos del **sistema existente se usan durante la especificación de requerimientos**.
 - Aclarar lo que hace el sistema existente. Discutir sus fortalezas y debilidades.
 - Conducen a los requerimientos para el nuevo sistema.
 2. **Modelos del sistema nuevo se emplean durante la especificación de requerimientos**.
 - Ayudar a explicar los requerimientos propuestos a otros participantes del sistema. Discutir las propuestas de diseño y documentar el sistema para la implementación.
 - En un proceso de ingeniería dirigido por modelo, es posible generar una implementación de sistema completa o parcial a partir del modelo del sistema.
- Un modelo del sistema es que deja fuera los detalles.
- **Es una abstracción** del sistema a estudiar, y no una representación alternativa de dicho sistema.

T4. Arquitecturas lógicas y físicas

- Puede desarrollar **diferentes modelos** para representar el sistema. Ejemplos:
 - Una **perspectiva externa**, donde se modelen el contexto o entorno del sistema.
 - Una **perspectiva de interacción**, donde se modele la interacción entre un sistema y su entorno, o entre los componentes.
 - Una **perspectiva estructural**, donde se modelen la organización de un sistema o la estructura de datos.
 - Una **perspectiva de comportamiento**, donde se modele el comportamiento dinámico del sistema y cómo responde ante ciertos eventos.
- Un estudio en 2007 (Erickson y Siau, 2007) mostró que la mayoría de los usuarios del UML consideraban que **cinco tipos de diagrama** podrían representar lo **esencial de un sistema**
 1. **Diagramas de actividad**, muestran las actividades incluidas en un proceso o en el procesamiento de datos.
 2. **Diagramas de caso de uso**, que exponen las **interacciones** entre un **sistema y su entorno**.
 3. **Diagramas de secuencia**, que muestran las **interacciones** entre los **actores y el sistema**, y entre los componentes del sistema.
 4. **Diagramas de clase**, que revelan las clases de objeto en el **sistema y las asociaciones** entre estas clases.
 5. **Diagramas de estado**, que explican cómo reacciona el sistema frente a eventos internos y externos.

T4. Arquitecturas lógicas y físicas

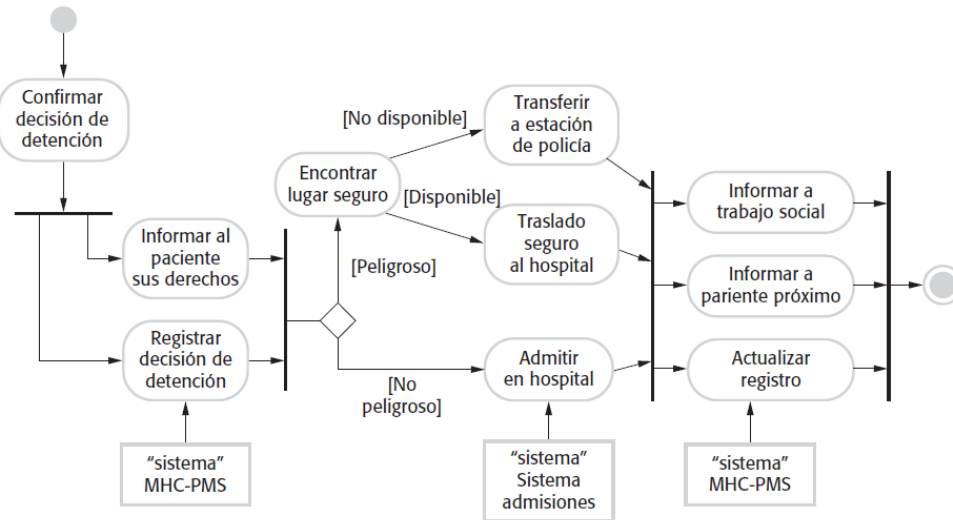


Diagrama de actividad

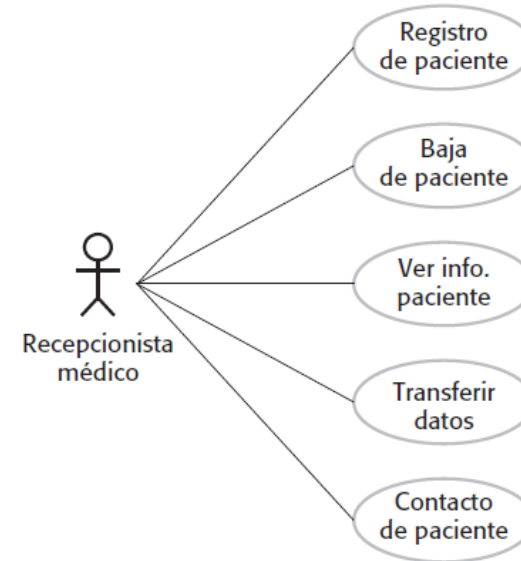


Diagrama de caso de uso

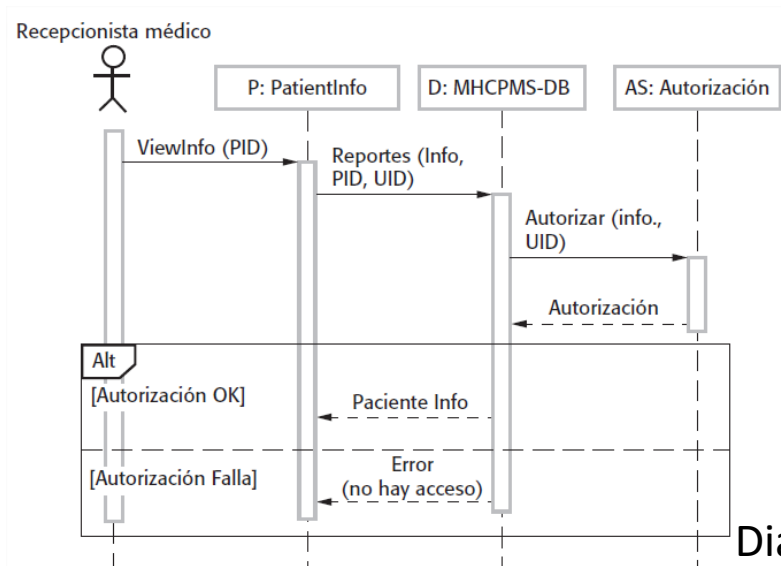


Diagrama de secuencia

T4. Arquitecturas lógicas y físicas

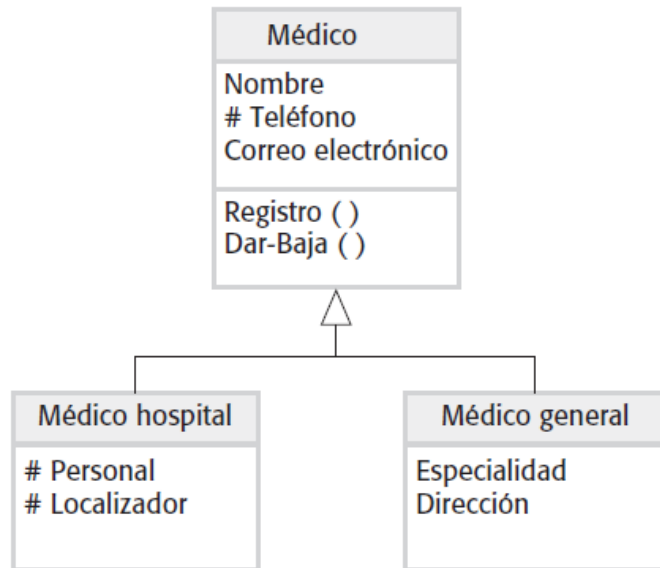


Diagrama de clase

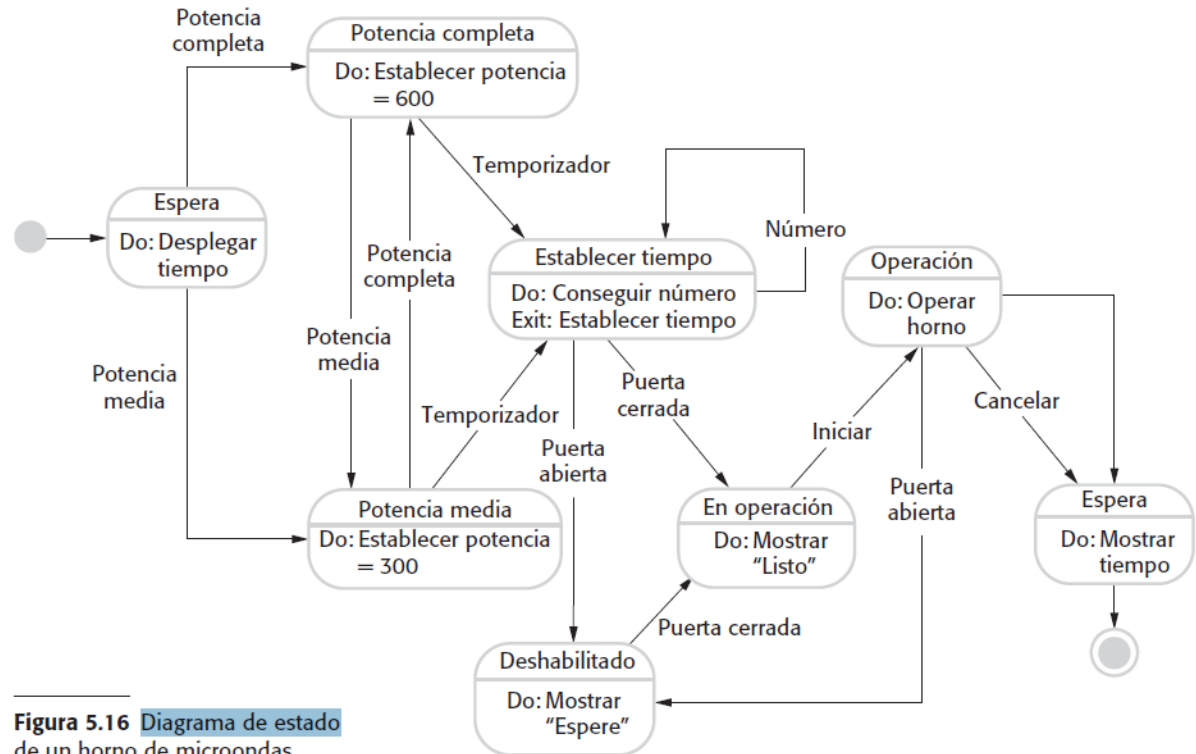


Figura 5.16 Diagrama de estado de un horno de microondas

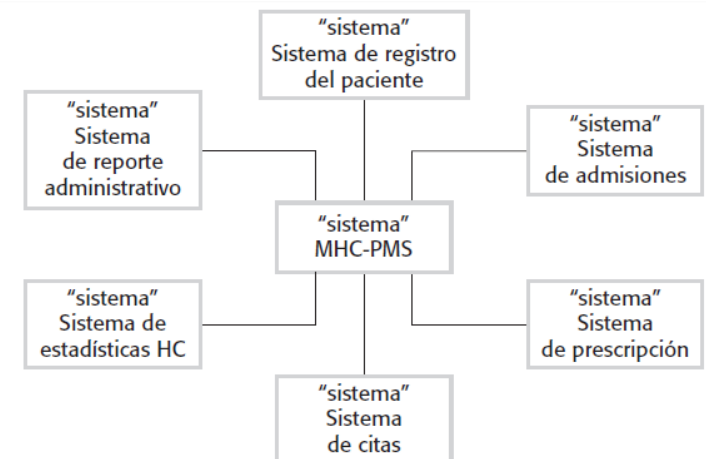
Diagrama de estado

T4. Arquitecturas lógicas y físicas. 4.1 Modelos de contexto

- Hay tres formas en que los modelos gráficos se emplean con frecuencia:
 1. Como medio para **facilitar la 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 generar una implementación de sistema.

4.1 Modelos de contexto

- La especificación de un sistema; debe decidir sobre las fronteras del sistema.
- **Trabajar con los participantes** del sistema para **determinar qué funcionalidad se incluirá en el sistema y qué ofrecerá el entorno del sistema**
- En algunos casos, la frontera entre un sistema y su entorno es relativamente clara.
 - Un sistema automático sustituye un sistema manual.
- En otros casos, existe más flexibilidad durante el proceso de ingeniería de requerimientos.
- Los modelos de contexto muestran que el entorno incluye sistemas automatizados.
- Éste es un modelo de contexto simple que muestra el sistema y otros sistemas en su entorno.



T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

- Los modelos de contexto simples se usan junto con otros modelos, como los modelos de proceso empresarial.
 - **Describen procesos humanos y automatizados** que se usan en sistemas particulares de software.

4.2 Modelos de interacción

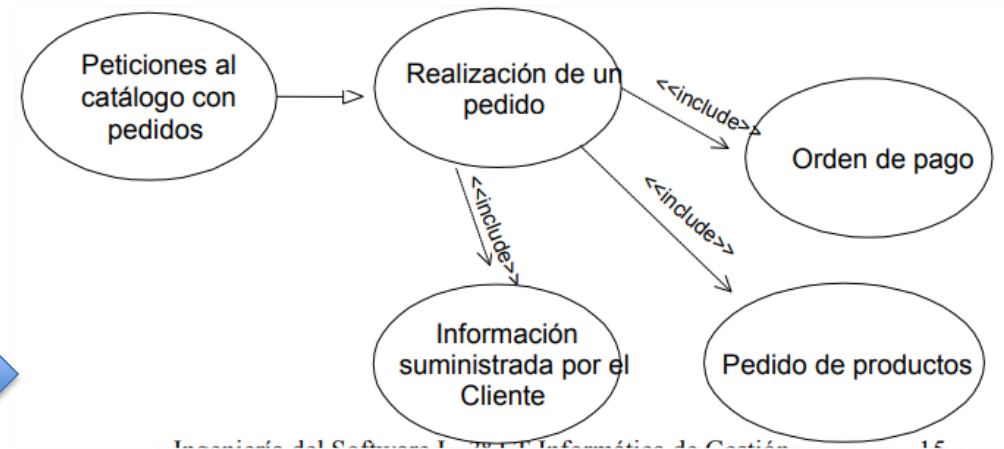
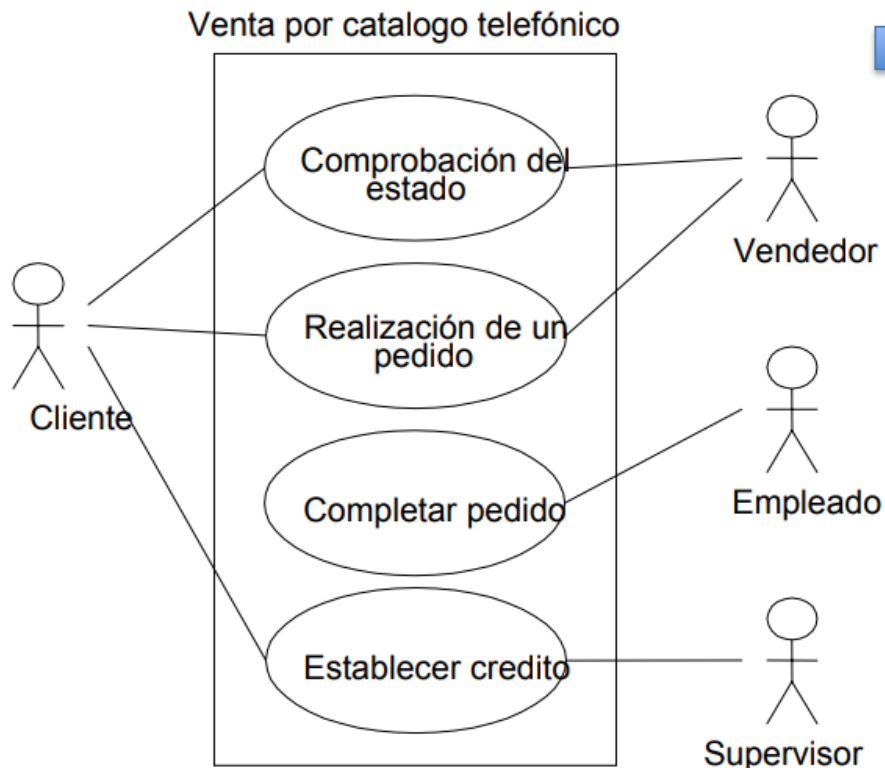
- Interacciones entre los sistemas: interacciones del usuario, que implican entradas y salidas del usuario; interacciones entre el sistema a desarrollar y otros sistemas; o interacciones entre los componentes del sistema.
- **Ayuda a identificar los requerimientos del usuario.**
- Enfoques del modelado de interacción:
 1. **Modelado de caso de uso**, que se utiliza principalmente para modelar interacciones entre un sistema y actores externos.
 2. **Diagramas de secuencia**, que se emplean para modelar interacciones entre componentes del sistema, o a agentes externos.

T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

1. Modelado de caso de uso

Herramientas:

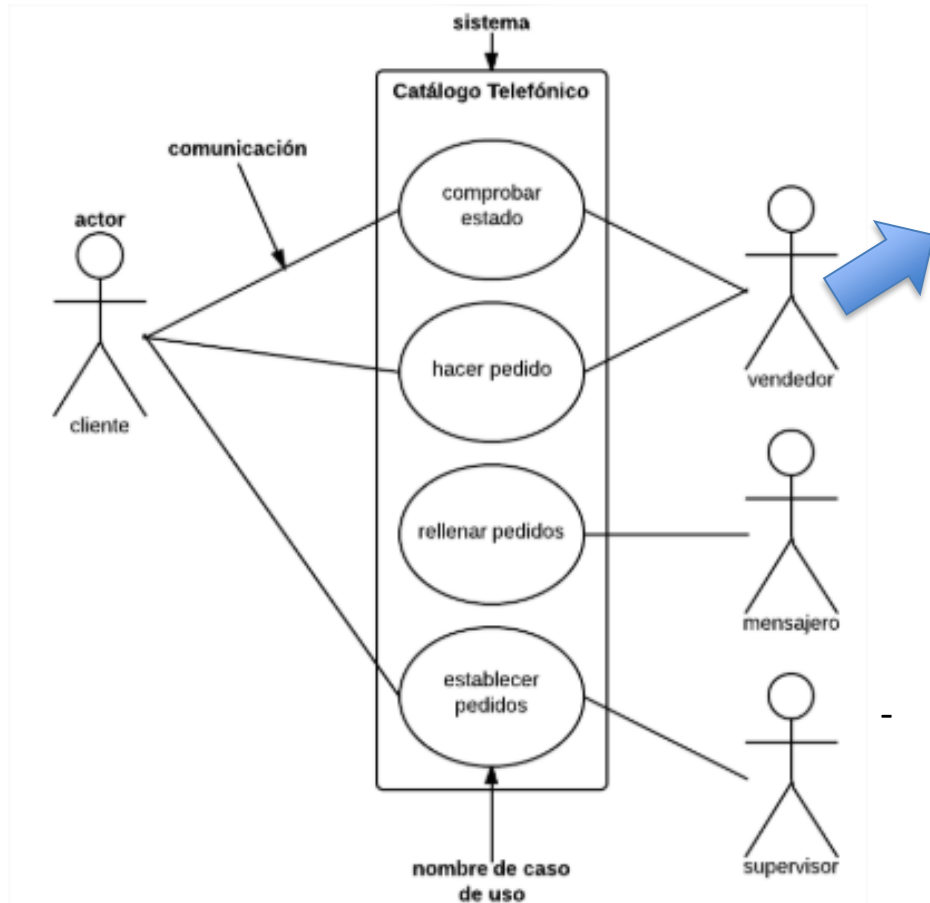
Papyrus UML (Eclipse), Draw.io, Lucidchart
(online y colaborativo)



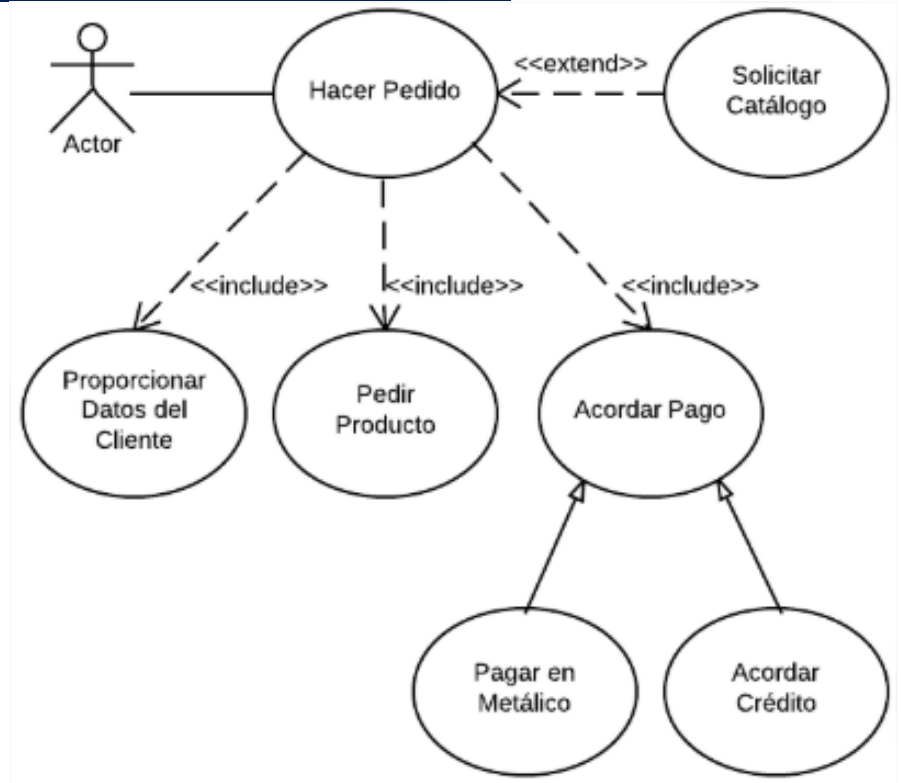
Relación	Función	Notación
Asociación	Camino de comunicación entre un actor y una caso de uso en el que participa	—————
Extiende	Inserción de comportamiento adicional en un caso de uso base (sin que éste tenga conocimiento)	—>>extiende>>—
Generalización	Relación entre un caso de uso general y otro más específico que hereda características y añade otras	—————>
Incluye	Inserción de comportamiento adicional dentro de un caso de uso que explícitamente describe la inserción	—>>incluye>>—

T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

Ejemplos

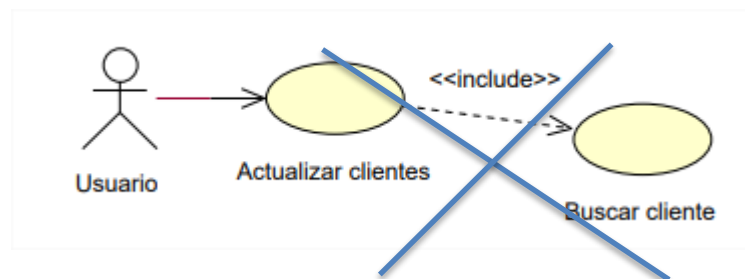
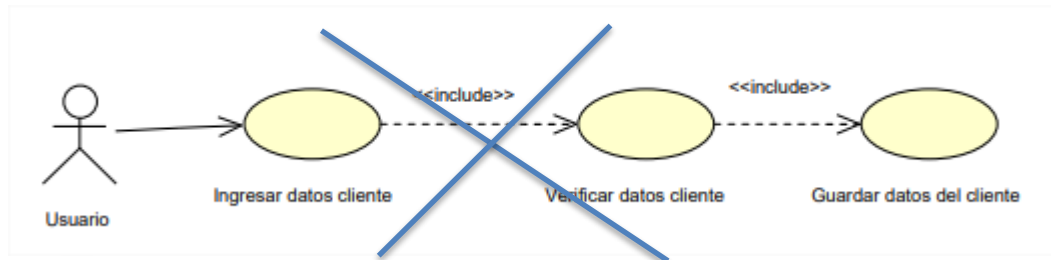


- Actor: cliente siempre ejecutará los casos de uso



- «Solicitar Catálogo» es una extensión del caso de uso «Hacer Pedido». El actor, cliente, puede solicitar un catálogo si lo desea cuando hace un pedido.
 - Se le ofrece y él elige.
- En el momento de acordar el pago, el actor puede pagar de cualquiera de las dos maneras. Se hace generalización para no duplicar relaciones, aunque se llamen desde distintos casos de uso.

T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción



Errores

- ¡Ojo! Los casos de uso son **funcionalidades** del sistema, acciones.
- No introducir: botones, listas desplegables, opciones de menú. Es especificación de requisitos, no diseño.
- No mencionar diseño de algoritmos o detalles de base de datos: “insertar en la tabla clientes”
- No escribir “así sucesivamente”...
- Que no parezcan diagramas de flujos de datos DFD.
- Que en un caso de uso sólo aparezca un «include», sin sentido completo en el contexto, sobre todo cuando está poco detallado.

T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

Comprobación del estado



<Identificador>	<nombre descriptivo>	
Descripción	El sistema deberá permitir a [lista actores] en [instante en el que se puede realizar el caso de uso] [funcionalidad que define el caso de uso] según se describe en el siguiente caso de uso:	
Secuencia Normal	Paso	Acción
	1	{<acción a realizar>, realizar el caso de uso [caso de uso]}
	2	<Situación que produce una alternativa>
	2a	Si [Situación que produce una alternativa] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}
	2b	Si [Situación que produce una alternativa] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}

	n
Excepciones	Paso	Acción
	p	En el caso de que [situación que provoca la excepción] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}

	q	...
Rendimiento	El sistema deberá realizar la/s acción /es descrita/s en {los pasos [primer paso] al [último paso], el paso [número de paso]} en un máximo de [cota de tiempo]	
Frecuencia	Este caso de uso se espera que se lleve a cabo una media de [número de veces] al [unidad temporal]	
Importancia	{vital, importante, quedaría bien}	
Urgencia	{inmediatamente, hay presión, puede esperar}	
Comentarios	<otras consideraciones en formato libre>	

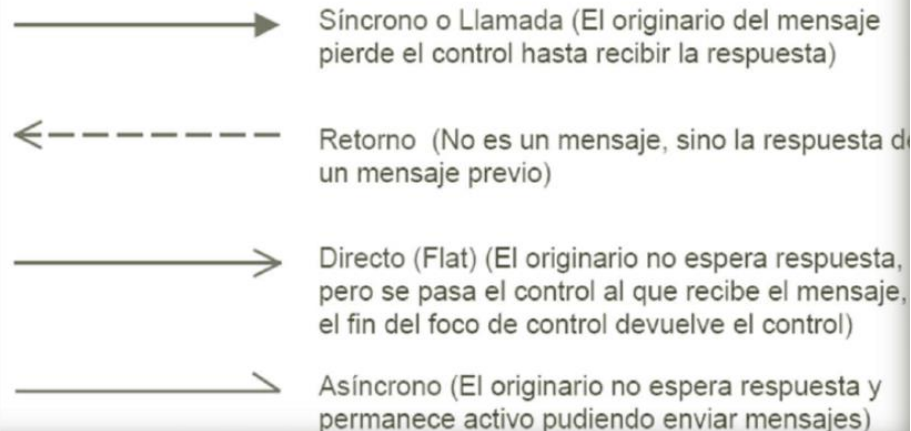
T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

2. Diagramas de secuencia

- **Un diagrama de secuencia** muestra la **sucesión de interacciones que ocurre durante un caso de uso particular.**
 - Su creación depende de la formulación previa de los casos de uso.
 - La secuencia de interacciones se lee de arriba abajo, el tiempo transcurre de esa forma.
 - Cuando existe retraso entre el envío y la recepción se puede indicar con una línea oblicua.
 - Muestra la secuencia de mensajes entre objetos durante un escenario concreto.
- **Línea de vida de un objeto** línea vertical punteada con un rectángulo. Ejecución de métodos (activación).
- **Activación** período de tiempo en el cual el objeto se encuentra desarrollando alguna operación. Rectángulo delgado sobre la línea de vida del objeto.
- **Mensaje** línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.
- **Tiempos de transición** En un entorno de objetos concurrentes o de demoras en la recepción de mensajes, es útil agregar nombres a los tiempos de salida y llegada de mensajes.
- **Caminos alternativos de ejecución y concurrencia** Estas alternativas pueden representar condiciones en la ejecución o diferentes hilos de ejecución.
- **Destrucción de un objeto** Se representa como una X al final de la línea de ejecución del objeto.

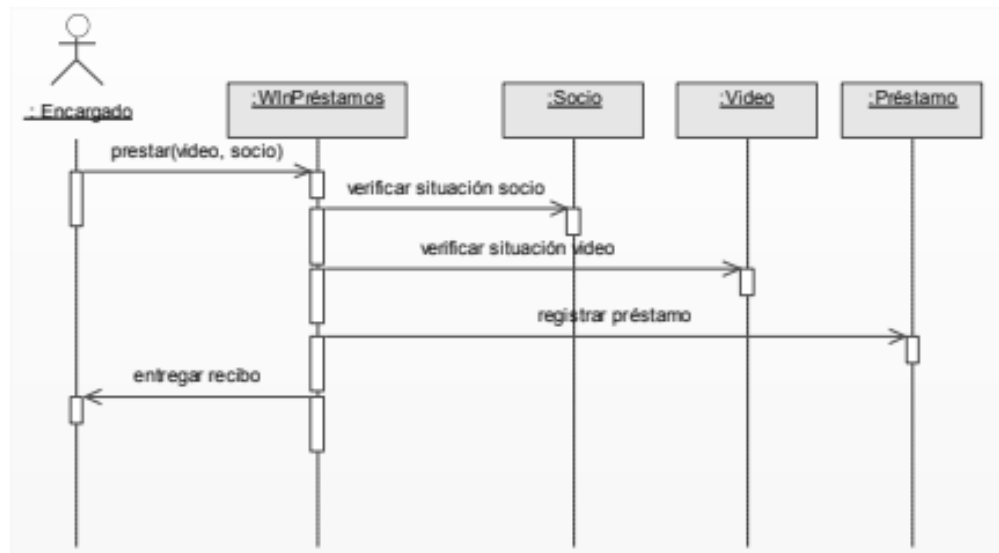
T4. Arquitecturas lógicas y físicas. 4.2 Modelos de interacción

Tipos de mensajes

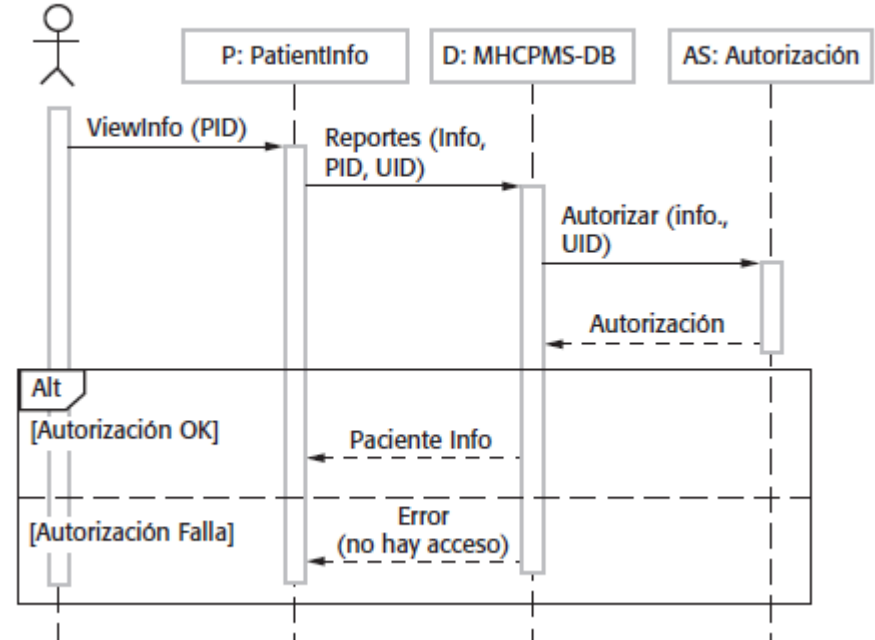


Tipos de fragmentos

Operador	Significado
alt	Indica que el fragmento de diagrama es una alternativa
loop	Indica que el fragmento de diagrama se ejecuta repetidas veces
opt	Indica que el fragmento de diagrama es opcional.
par	Indica que el fragmento de diagrama incluye varias hebras



Recepcionista médico



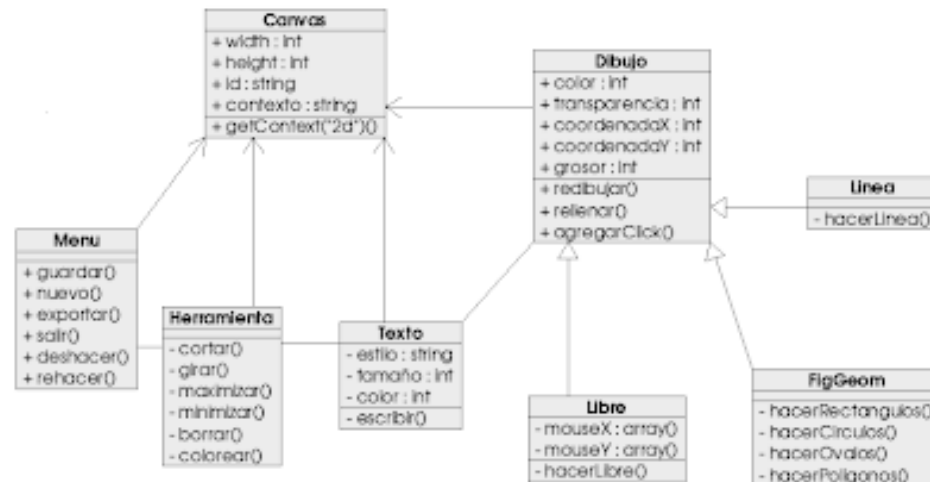
T4. Arquitecturas lógicas y físicas. 4.3 Modelos estructurales

- Los modelos estructurales son modelos estáticos, que muestran la estructura del diseño del sistema, o modelos dinámicos, que revelan la organización del sistema cuando se ejecuta.
- Los modelos estructurales de un sistema se crean cuando se discute y diseña la arquitectura del sistema.

4.3.1 Diagramas de clase

- Pueden usarse cuando se desarrolla un **modelo de sistema orientado a objetos para mostrar las clases en un sistema** y las asociaciones entre dichas clases.
- Se desarrollan modelos durante **las primeras etapas del proceso de ingeniería de software**. el enfoque está sobre el modelado de objetos del mundo real, como parte de los requerimientos o los primeros procesos de diseño del software.
- La forma más sencilla de hacer esto es escribir el nombre de la clase en un recuadro. También puede anotar la existencia de una asociación dibujando simplemente una línea entre las clases

Es posible usar el UML para representar un modelo semántico de datos aunque no incluya una notación específica



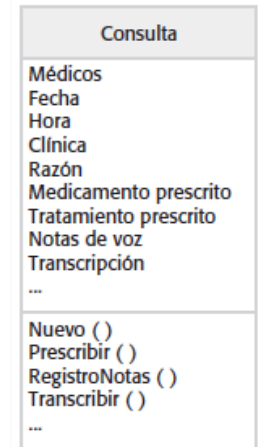
T4. Arquitecturas lógicas y físicas. 4.3 Modelos estructurales



Universidad
Francisco de Vitoria
UFV Madrid

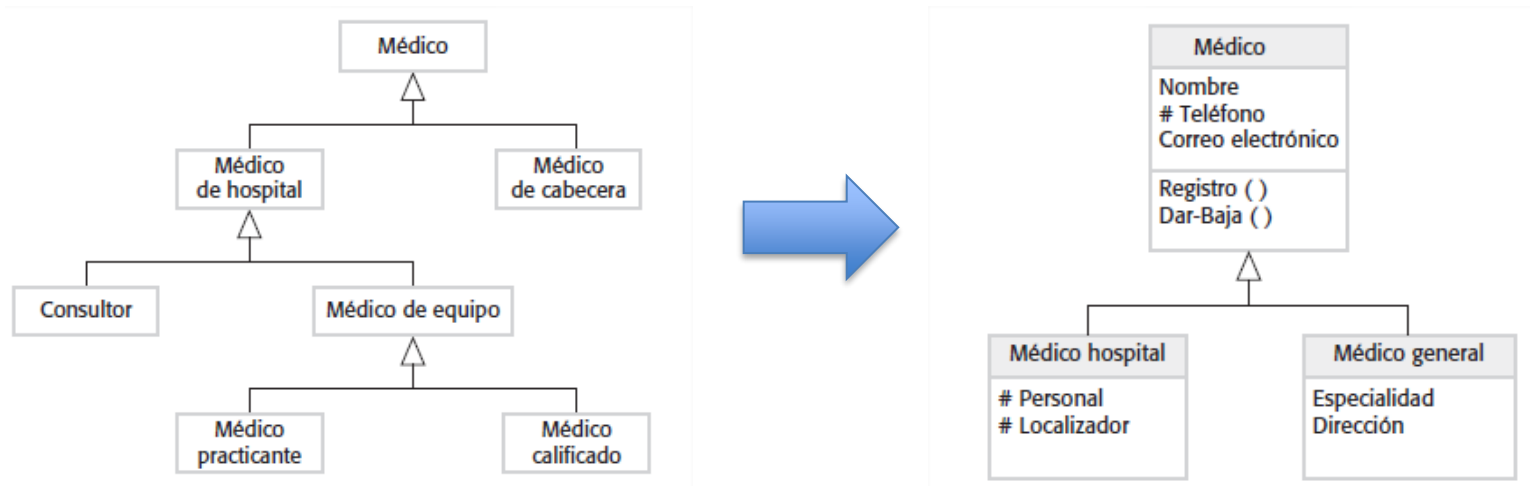
4.3.1 Diagramas de clase

1. El nombre de la clase de objeto está en la sección superior.
2. Los atributos de clase están en la sección media. Esto debe incluir los nombres del atributo y, opcionalmente, sus tipos.
3. Las operaciones.



4.3.2 Generalización

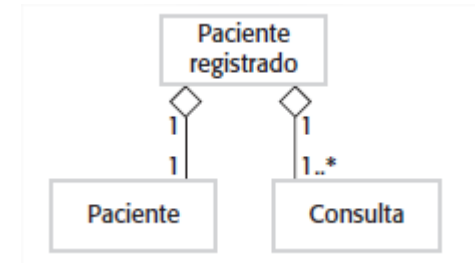
- Técnica cotidiana que se usa para gestionar la complejidad. Las entidades se colocan en clases más generales y se agrupan las características comunes. Herencia.



T4. Arquitecturas lógicas y físicas. 4.4 Modelos de comportamiento

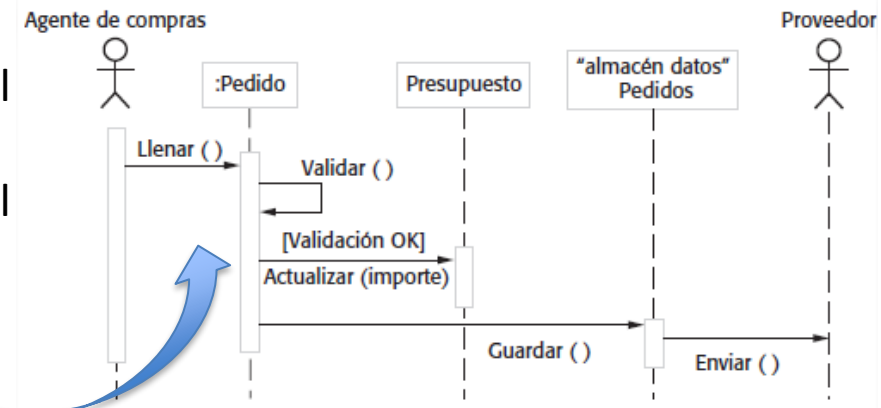
4.3.3 Agregación

- El UML proporciona un tipo especial de asociación entre clases llamado agregación, que significa que **un objeto (el todo) se compone de otros objetos (las partes)**. Para mostrarlo, se usa un trazo en forma de diamante, junto con la clase que representa el todo



4.4. Modelos de comportamiento

- Son modelos dinámicos del sistema conforme se ejecutan. En ellos se muestra lo que sucede o lo que se supone que pasa cuando un sistema responde ante un estímulo de su entorno.
 - 1. *Datos* - datos que llegan se procesan por el sistema.
 - 2. *Eventos* - eventos activan el procesamiento del sistema. A veces llevan datos.
- Modelado **dirigido por datos** (d. secuencia).
- Modelado dirigido por un evento (d. estado).



T4. Arquitecturas lógicas y físicas. 4.4 Modelos de comportamiento

- Modelado dirigido por un evento

Ejemplo: un sistema que controla una válvula puede moverse de un estado de “válvula abierta” a un estado de “válvula cerrada”, cuando recibe un comando operador (el estímulo)

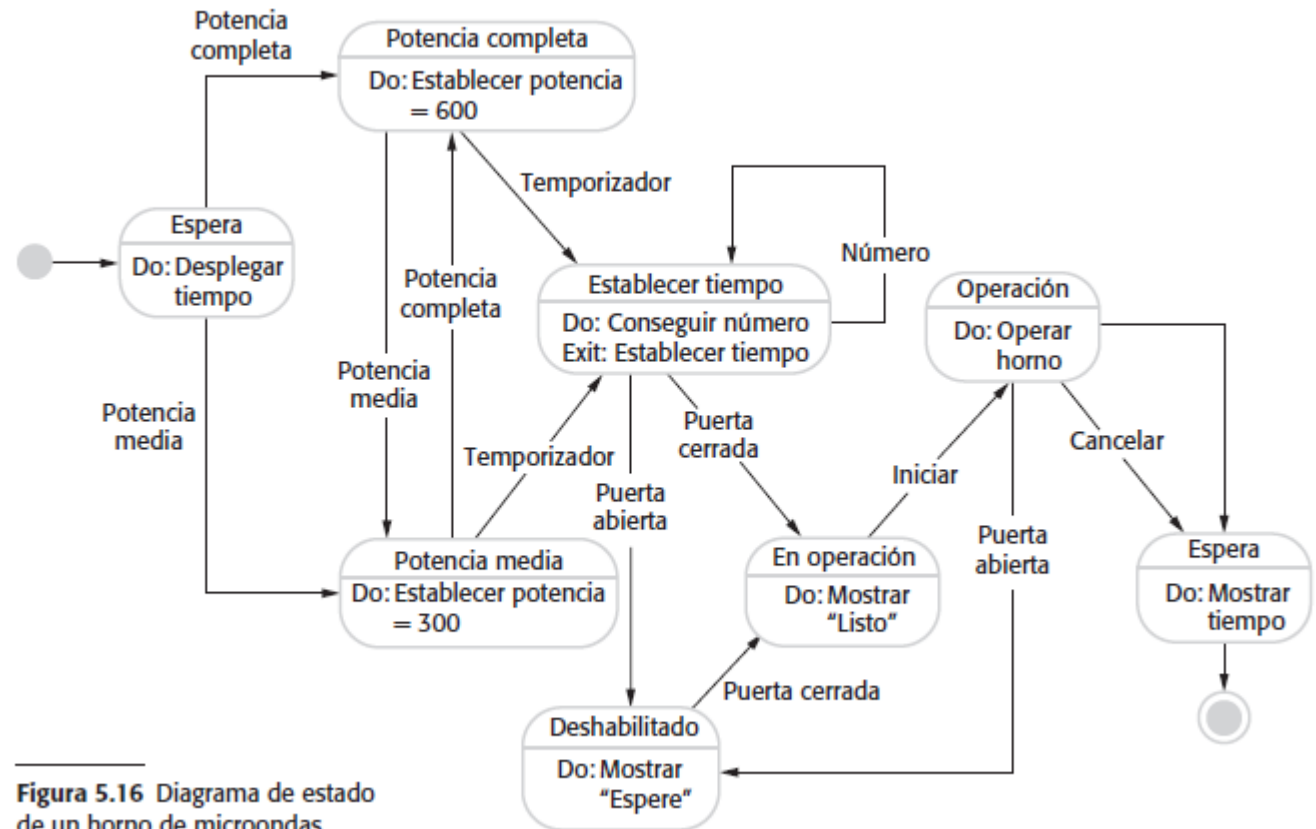


Figura 5.16 Diagrama de estado de un horno de microondas

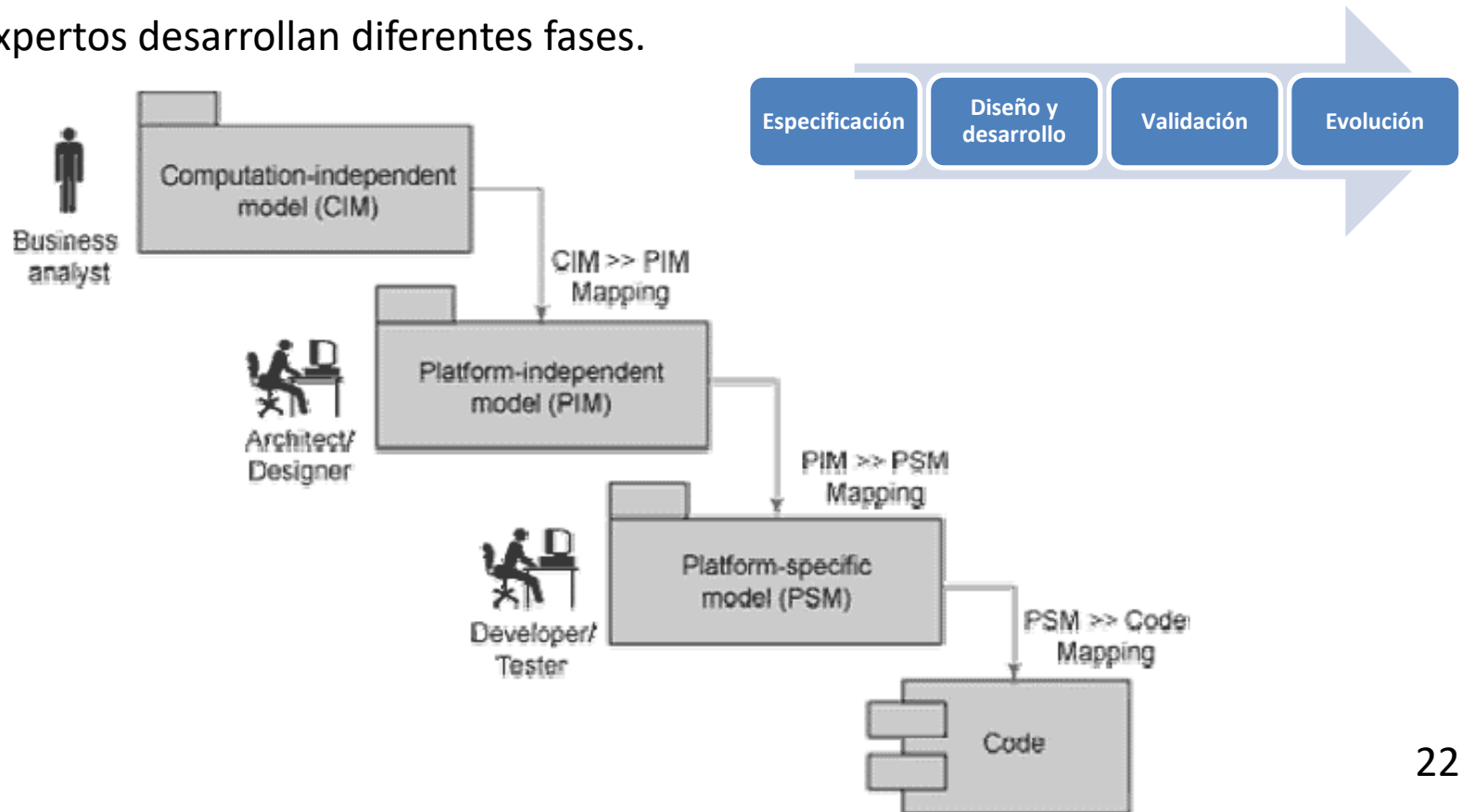
T4. Arquitecturas lógicas y físicas. 4.5 Ingeniería dirigida por modelo

- Un enfoque al desarrollo de software **donde los modelos, y no los programas**, son las **salidas principales** del proceso de desarrollo (Kent, 2002; Schmidt, 2006).
- **Los programas** que se ejecutan en una plataforma hardware/software se **generan automáticamente a partir de los modelos**.
- **Arquitectura dirigida por modelos**
 - **La MDA** (*Model-Driven Architecture*) se enfoca en las etapas de **diseño e implementación del desarrollo de software**, mientras que **la MDE** (*Model-Driven Engineering*) se interesa por **todos los aspectos del proceso de ingeniería de software**.
 - Enfoque promovido por (OMG) Object Management Group en 2001.
 - Es un enfoque de desarrollo de software a través del uso de modelos previos al código.
 - **Principios:**
 - **Representación directa:** esta estrategia se basa en el principio de abstracción, que hace énfasis en el dominio del problema más que en la tecnología.
 - **Automatización:** surgen nuevas funcionalidades que deben ser soportadas por las herramientas como el intercambio de modelos.
 - **Estándares abiertos:** el uso de estándares se ha constituido en el medio que ha posibilitado el reto de integrar herramientas robustas de apoyo al desarrollo.

T4. Arquitecturas lógicas y físicas. 4.5 Ingeniería dirigida por modelo

Ventajas de MDA

- Menor esfuerzo de desarrollo ya que parte de ello lo genera los modelos PIM (modelos resultantes del análisis y diseño de un sistema concreto).
- Portabilidad e independencia de la plataforma.
- Aumento el nivel de abstracción.
- Mayor facilidad de mantenimiento.
- Distintos expertos desarrollan diferentes fases.



T4. Arquitecturas lógicas y físicas.

4.5 Ingeniería dirigida por modelo

Arquitectura dirigida por modelos

- La clave de MDA es la importancia que le da a los modelos en los procesos de desarrollo.
- Tres módulos de desarrollo:

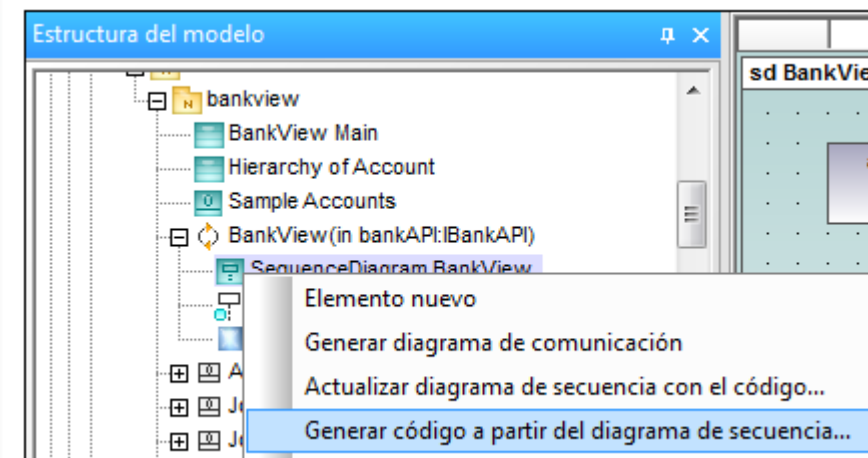
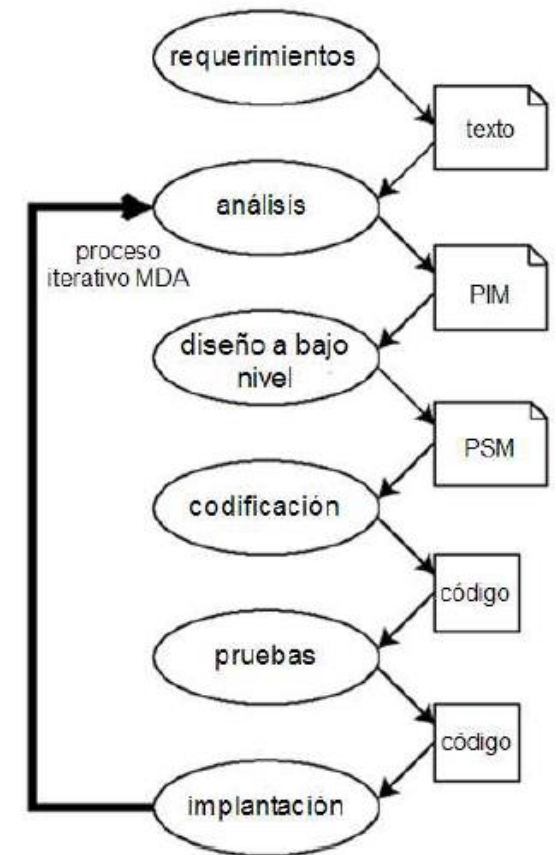
Modelos independientes del computo (CIM):
semejante a los modelos del dominio y/o del negocio del proceso unificado de desarrollo.

Modelos independientes de la plataforma (PIM):
son modelos con un alto nivel de abstracción que son independientes de la tecnología en la que se van a implantar.

Modelos específicos de plataforma (PSM):
especifican el sistema en términos de las construcciones que se van a implementar a la hora de desarrollar.



Código: la fase final del desarrollo es transformar cada PSM (modelo específico de la plataforma) en código.



T4. Arquitecturas lógicas y físicas. 4.5 Ingeniería dirigida por modelo

- **Ventajas MDE(Engineering):** La ingeniería basada en modelo permite a los ingenieros pensar sobre sistemas en un nivel de abstracción elevado, sin ocuparse por los detalles de su implementación.
 - **Reduce errores.**
 - **Acelera el diseño.**
 - **Reutilización con independencia de la plataforma.**
- **Desventajas MDE:** **no siempre** se sigue que las abstracciones que soporta el modelo **son las abstracciones correctas para la implementación**. Los argumentos para independencia de plataforma **sólo son válidos para sistemas grandes de larga duración**, donde las plataformas se vuelven obsoletas durante el tiempo de vida de un sistema.
 - **No todas las abstracciones de diseño son válidas para implementación.**