

Unidad III

Tema: Modelado del Contexto y Estructura del Sistema con UML

Aspectos que se modelan de un Sistema

- Contexto
- Estructura
- Dinámica (Interacción)
- Comportamiento

Modelos de Contexto de un Sistema

Modelos de contexto del sistema

En una primera etapa en la especificación de un sistema, debe decidir sobre las *fronteras del sistema*.

Esto implica trabajar con los participantes del sistema para determinar **cuál funcionalidad se incluirá en el sistema y cuál la ofrece el entorno del sistema**.

El apoyo automatizado para algunos procesos puede o debe implementarse, pero otros deben ser procesos manuales o soportados por sistemas diferentes.

Estas decisiones deben hacerse durante el proceso, para limitar los costos del sistema, así como el tiempo necesario para comprender los requerimientos y el diseño del sistema

Modelo de contexto. Ejemplo Sistema MHC-PMS.

Ejemplo: Especificación para el sistema de información de pacientes para atención a la salud mental **MHC-PMS**.

Objetivo: Este sistema intenta manejar la información sobre los pacientes que asisten a clínicas de salud mental y los tratamientos que les prescriben.

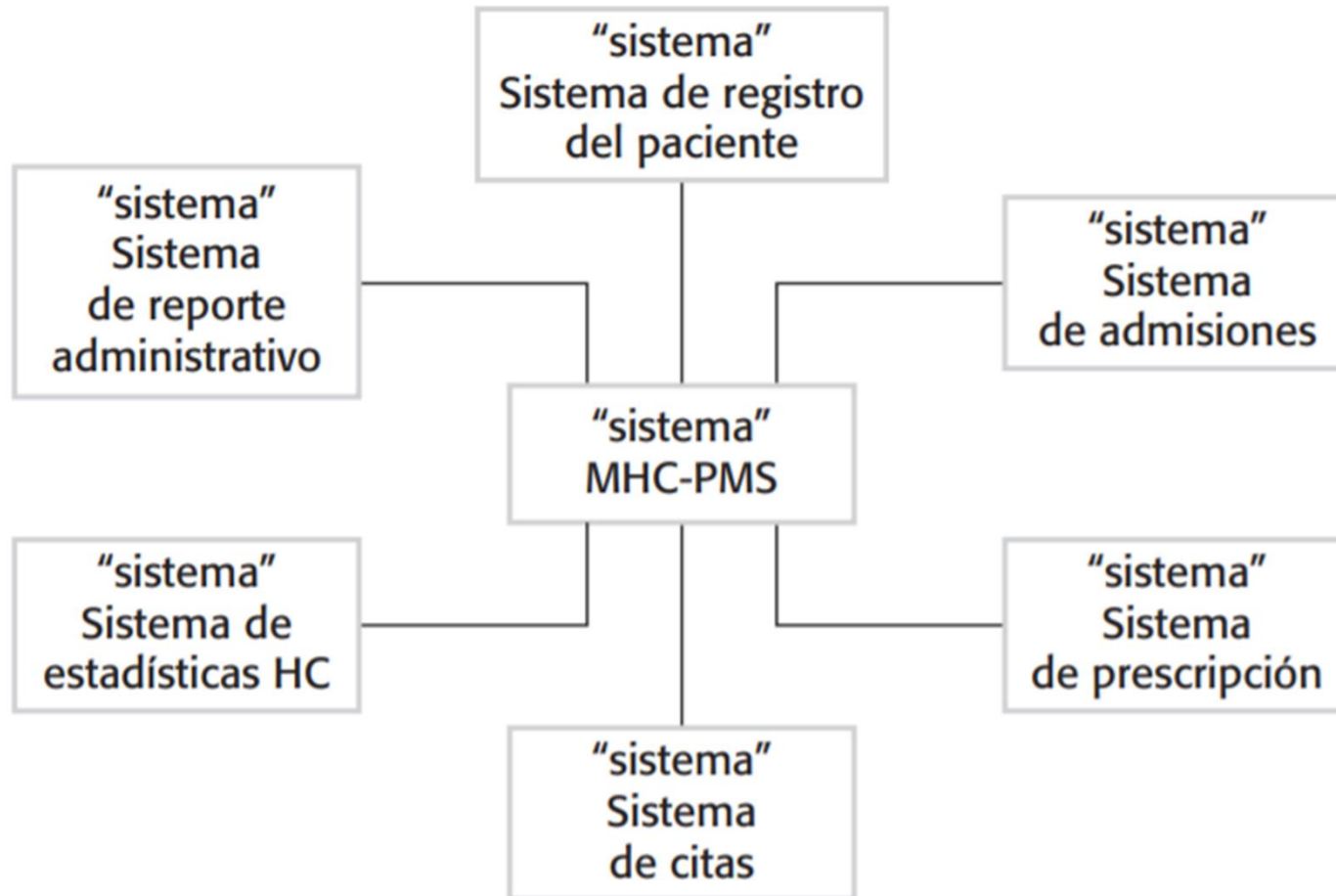
Modelo de contexto. Ejemplo Sistema MHC-PMS.

Cuestiones de la Especificación:

- La especificación para este sistema, debe *decidir si el sistema tiene que enfocarse exclusivamente en reunir información de las consultas, o si también es necesario que recopile datos personales* acerca del paciente.
- La ventaja de apoyarse en otros sistemas para la información del paciente es que evita duplicar datos.
- Sin embargo, la principal desventaja es que usar otros sistemas haría más lento el acceso a la información.
- Si estos sistemas no están disponibles, entonces no pueden usarse en MHC-PMS.

Modelo de contexto. Ejemplo Sistema MHC-PMS.

Es un modelo de contexto simple que muestra el sistema de información del paciente y otros sistemas en su entorno



Fuente ejemplo, Sommerville

Modelo de contexto. Ejemplo Sistema MHC-PMS

El MHC-PMS está conectado con:

- un sistema de citas y un sistema más general de registro de pacientes, con el cual comparte datos.
- sistemas para manejo de reportes y asignación de camas de hospital, y un sistema de estadísticas que recopila información para la investigación.
- utiliza un sistema de prescripción que elabora recetas para la medicación de los pacientes

Modelo de contexto. Restricciones

- Los modelos de contexto, muestran que el entorno incluye varios sistemas automatizados.
- Sin embargo, no presentan los tipos de relaciones entre el sistema y los sistemas en el entorno
 - ✓ Los sistemas externos generan datos para el sistema o consumen datos del sistema.
 - ✓ Pueden compartir datos con el sistema, conectarse directamente, a través de una red, o no conectarse.
 - ✓ Pueden estar físicamente juntos o ubicados en edificios separados.
 - ✓ Todas estas relaciones deben tomarse en cuenta en la especificación de requerimientos.
 - ✓ los modelos de contexto simples se usan junto con otros modelos, como los **modelos de negocio**.

Modelo de contexto. RUP

- RUP propone modelar el Contexto de un Sistema, según dos modalidades:
- Construcción de un **Modelo de Dominio**
 - Incluyendo Modelo de objetos del dominio
- Construcción de un **Modelo de Negocio**
 - Incluyendo Objetos del Negocio y Casos de Uso del negocio

Modelos de estructura del Sistema

Diagrama de clases y objetos

Diagrama de Packages

Modelos estructurales

- Los modelos estructurales de software muestran la organización de un sistema, en términos de los componentes que constituyen dicho sistema y sus relaciones.
- 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.

Modelos estructurales

Los modelos estructurales de software muestran la **organización de un sistema**, en términos de los componentes que constituyen dicho sistema y sus relaciones

Los modelos estructurales ***son modelos estáticos***, que muestran la estructura del diseño del sistema

Los modelos estructurales ***son 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.

Diagrama de clases

- Los **diagramas de clase** pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar:



Clases en un sistema y las asociaciones entre dichas clases.

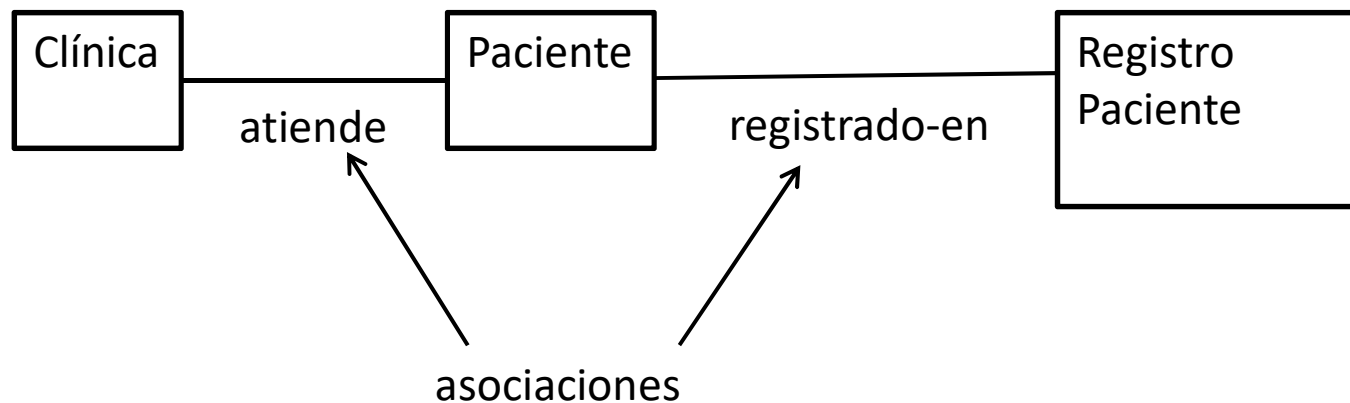
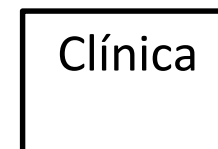


Diagrama de clases

- Durante las primeras etapas de Modelado del SW, el enfoque está puesto en modelar los objetos representan algo *en el mundo real*, como un paciente, una receta, un médico, etc.
 - ✓ La primera etapa consiste en buscar conceptos en el mundo real, identificar los objetos esenciales y representarlos como clases
 - ✓ No son Objetos SW!!!! Todavía no tienen comportamiento (métodos)






Es una clase conceptual
No es Objeto SW

Diagrama de clases

- El Diagrama de clases forma parte de la vista estática del sistema
 - Es una representación gráfica de un modelo estructural
 - Describe los tipos de objetos de un sistema y las diversas relaciones estáticas que existen entre ellos.
 - Se usa para modelar las abstracciones relevantes del dominio del problema.

Perspectivas del Diagrama de clases:

Niveles clases de Fowler

- **Conceptual** 
 - Se dibuja un diagrama que represente los conceptos del dominio que se está estudiando (no interesa el SW)
- **Especificación** 
 - Se ven las interfaces del software, es decir atributos y operaciones (pueden tener diferentes implementaciones).
- **Implementación** 
 - Se expone por completo la implementación de las clases: de atributos y métodos

Operaciones en diferentes niveles del DC

- **Nivel conceptual:**

- Responsabilidades de la clase
- Ej. en tarjetas CRC: Descripción de alto nivel del propósito de la clase

- **Nivel especificación:**

- Comportamiento (protocolo) de la clase:
Operaciones públicas

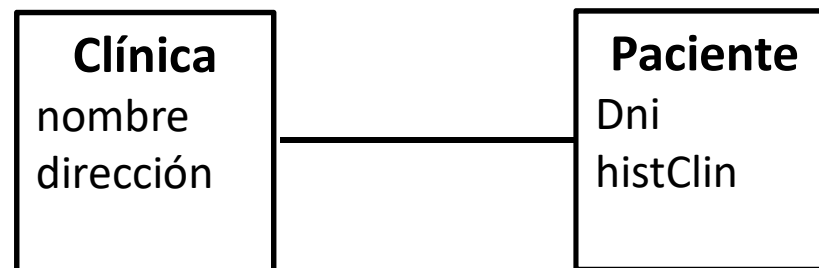
- **Nivel Implementación**

- Conjunto de métodos de la clase.

Perspectivas del Diagrama de clases:

Niveles clases de Fowler: Dónde estamos en Modelado???

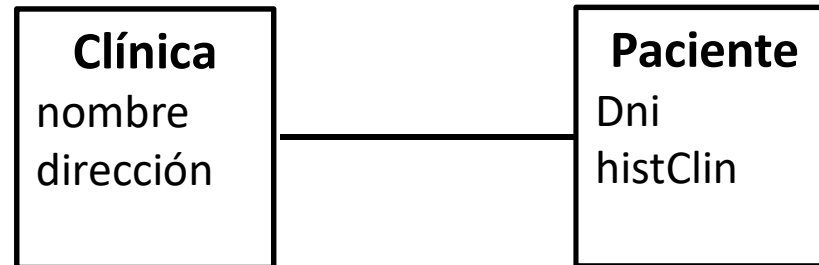
- **Conceptual**
- Se dibuja un diagrama que represente los conceptos del dominio que se está estudiando (no interesa el SW)
- *A lo sumo se pueden definir algunos atributos en este nivel*



Clases de nivel conceptual

Perspectivas del Diagrama de clases:

Niveles clases de Fowler: Dónde estamos en Modelado???



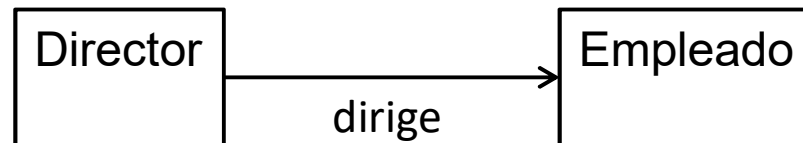
- **Nivel Conceptual:** “La Clínica tiene un nombre”. “El paciente tiene un dni”
- **Nivel Especificación:** “El cliente puede almacenar y consultar su dni”. “La Clínica puede modificar su dirección”
- **Nivel Implementación:** “La clínica tiene un campo de tipo Srting que almacena su nombre y un método que lo devuelve”

Relaciones entre clases

- Las clases tienen relaciones con otras clases
- A nivel conceptual, las clases presentan las siguientes relaciones:
 - Asociación
 - Generalización/especialización
 - Agregación
 - Composición

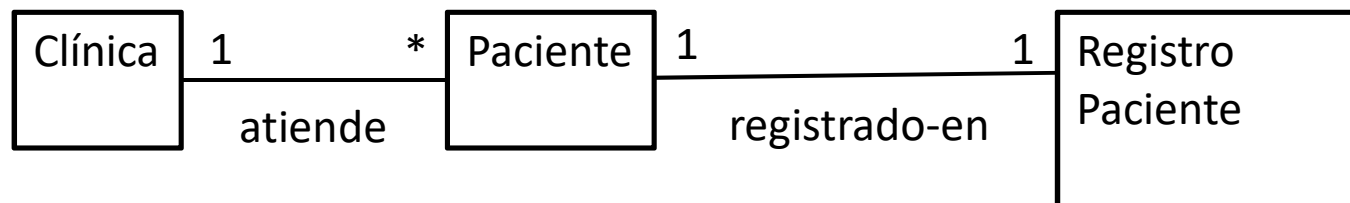
Relaciones ente clases. Asociación

- Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos.
- A una asociación se le puede asignar un nombre
- La asociación se representa en UML con una línea continua, **posiblemente dirigida**, que a veces incluye una etiqueta



Asociación. Multiplicidad

- En los Diagramas conceptuales se puede mostrar cuántos objetos intervienen en la asociación
- Por ejemplo, una Clínica puede tener muchos pacientes, pero cada paciente tiene un solo registro (historia clínica)
- Esta propiedad se llama “multiplicidad” y puede mostrarse en este nivel de modelado de clases.

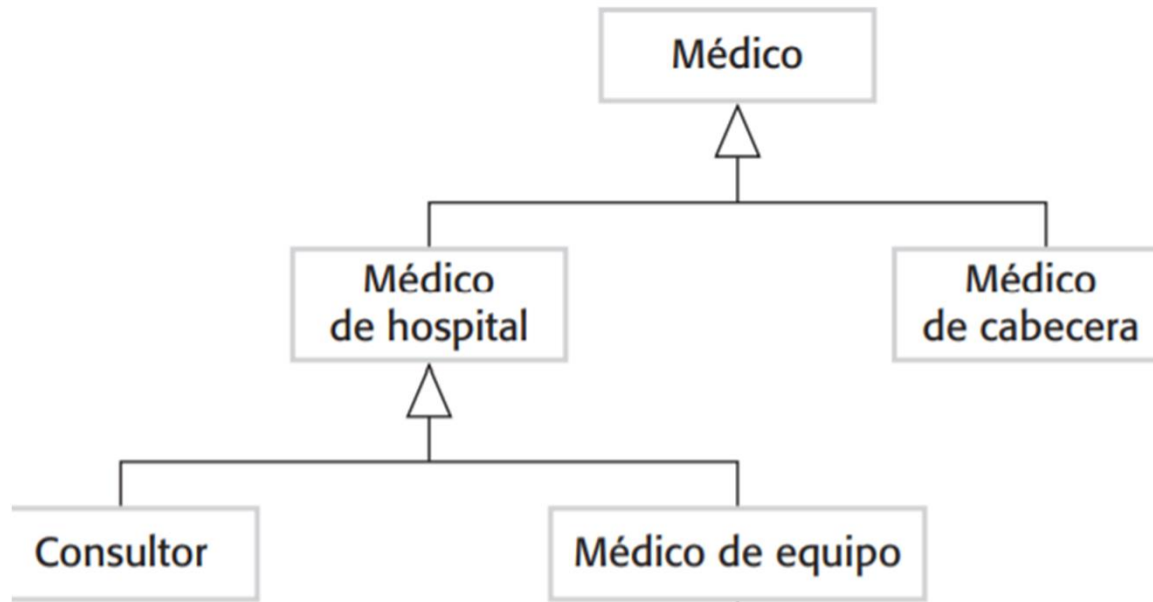


Relaciones entre clases. Generalización

- La generalización es una técnica que se usa para gestionar la **complejidad**. Permite modelar jerarquías de clases
- En el modelado de sistemas, con frecuencia es útil examinar las clases en un sistema, con la finalidad de ver si hay lugar para la generalización.
- En vez de modelar las características detalladas de cada entidad, dichas entidades se colocan en clases más generales (animales, automóviles, casas, etc.)
 - Diferentes miembros de estas clases tienen algunas características comunes (por ejemplo, las ardillas y ratas son roedores).
- Esto significa que la información común se mantendrá solamente en un lugar.

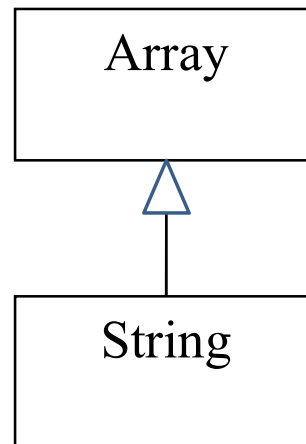
Generalización en UML

- En UML hay una técnica para representar esta relación entre clases.
- Se muestran los distintos niveles jerárquicos



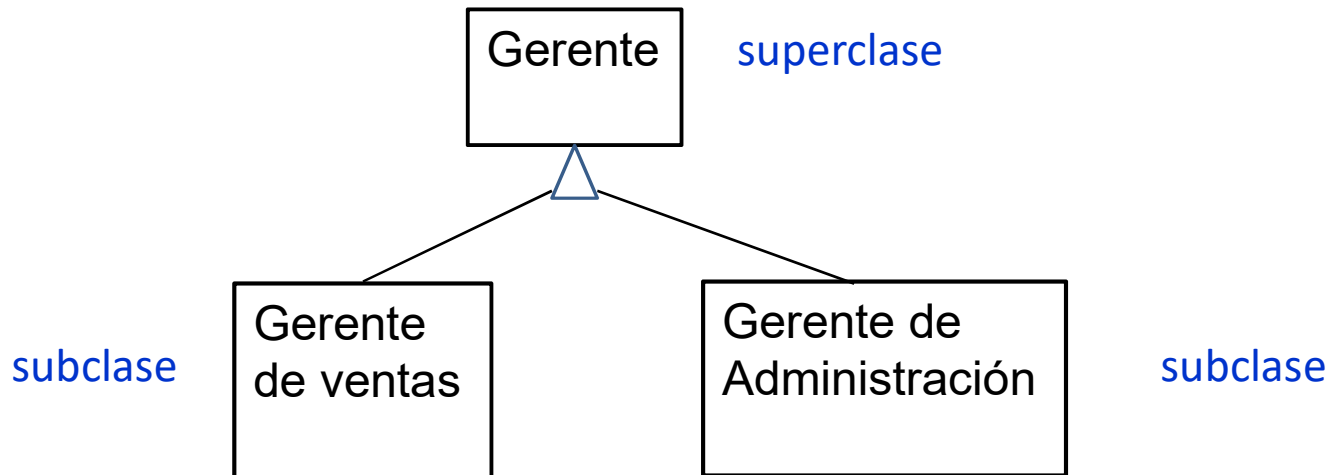
Generalización. Herencia

- La generalización se logra a través de un mecanismo de abstracción de OO: Herencia
- Ej.: Un string se comporta como un array, porque sus elementos pueden accederse por un índice. Sin embargo tienen características adicionales, por ej. ponerse en orden alfabético



Generalización. Superclases y subclases

- Una superclase es una clase de la cual se heredan sus atributos y su comportamiento.
- Una clase puede tener una o varias superclases, combinando comportamiento de distintas fuentes y agregando sólo un poco de si misma para producir su propio tipo de objeto.



Generalización. Atributos y operaciones

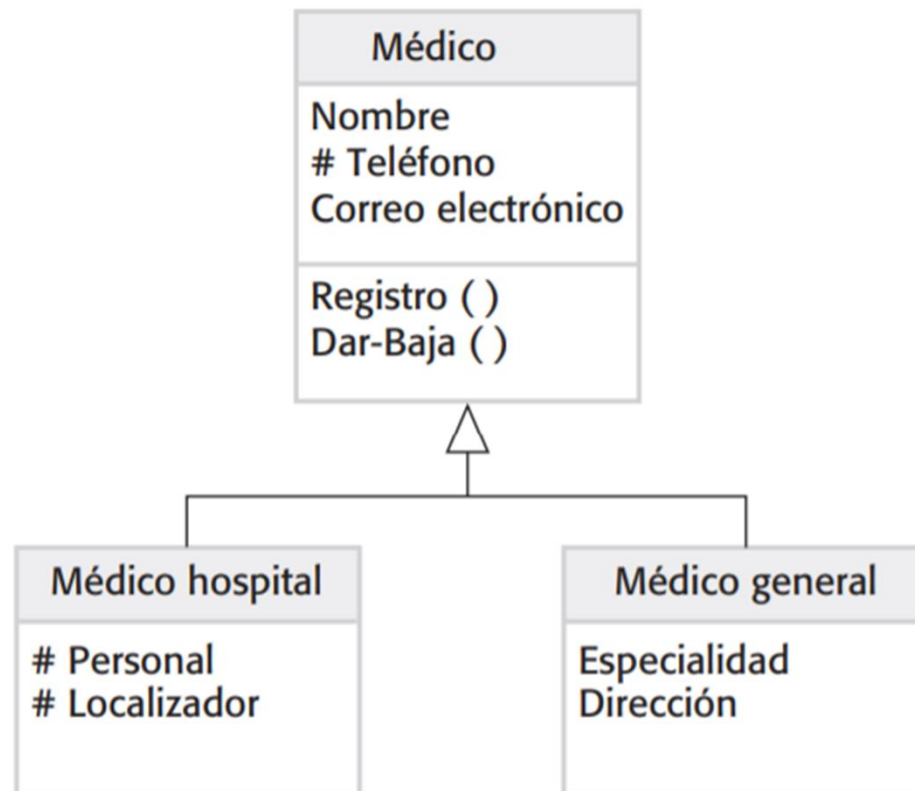
- En una generalización, los atributos y las operaciones asociados con las clases de nivel superior también se asocian con las clases de nivel inferior.



- En esencia, las clases de nivel inferior son **subclases que heredan los atributos y las operaciones** de sus superclases.
- Dichas clases de nivel inferior **agregan atributos y operaciones** más específicos
 - Por ejemplo, todos los médicos tienen un nombre y número telefónico; todos los médicos de hospital tienen un número de personal y un departamento, pero los médicos de cabecera no tienen tales atributos, pues trabajan de manera independiente. Sin embargo, sí tienen un nombre de consultorio y dirección.

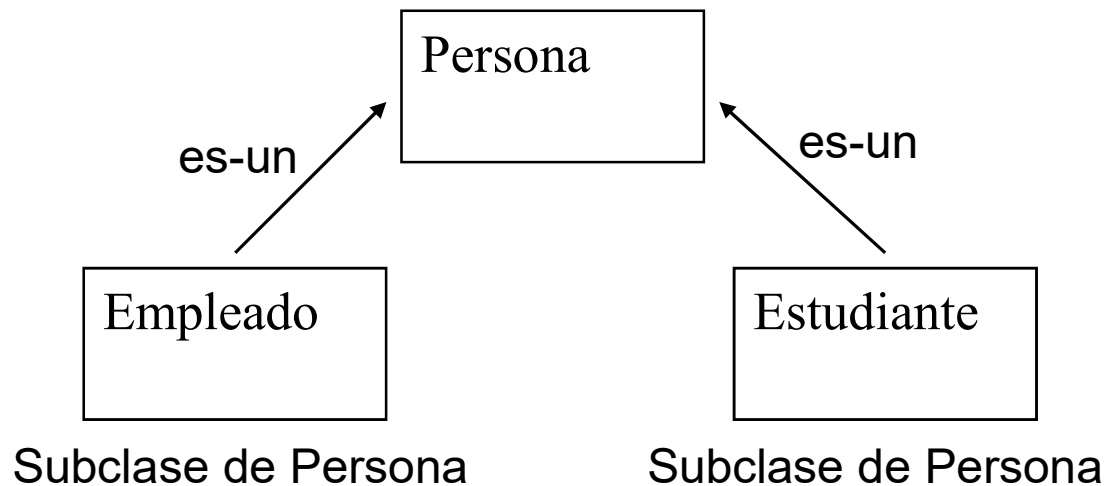
Generalización. Atributos y operaciones

- Por ejemplo, todos los médicos tienen un nombre y número telefónico; todos los médicos de hospital tienen un número de personal y un localizador, pero los médicos de cabecera no tienen tales atributos, pues trabajan de manera independiente. Sin embargo, sí tienen un nombre de consultorio y dirección.



Generalización. Jerarquías de clases

- El *mecanismo de abstracción* de la generalización / especialización, consiste en construir:
 - *jerarquías de clase / subclase*, donde, en general las clases están vinculadas por la relación es-un.
- Por ej., describimos la clase empleado como sub-clase de persona (un empleado es-una persona). La clase estudiante es sub-clase de persona.

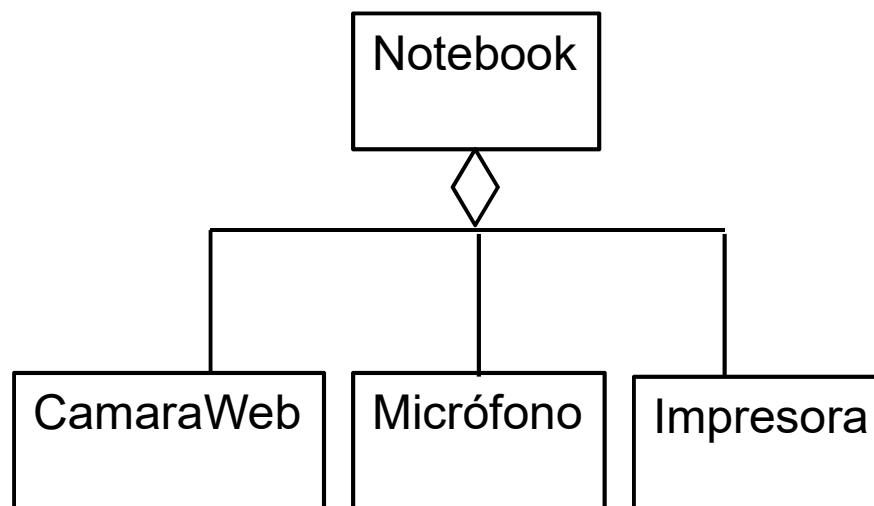


Relaciones entre clases. Agregación.

- Los objetos en el mundo real con frecuencia están compuestos por diferentes partes.
- Un paquete de estudio para un curso, por ejemplo, estaría compuesto por libro, diapositivas de PowerPoint, exámenes y recomendaciones para lecturas posteriores.
- En un modelo de sistema, normalmente se necesita ilustrar esto

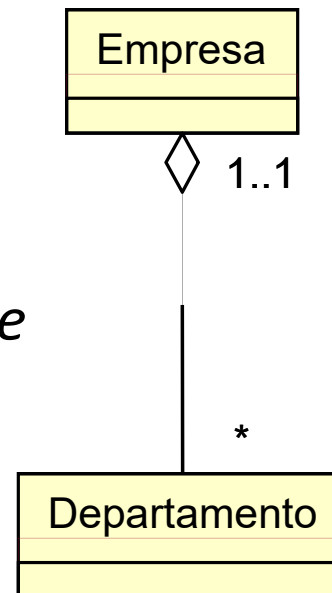
Agregación en UML.

- Por ejemplo, a una computadora portátil se le pueden agregar elementos como micrófono, cámara web e impresora; sin embargo la ausencia de estos elementos no repercute en el funcionamiento básico de dicha portátil.
- El símbolo de agregación es un diamante vacío colocado en el extremo de la clase que contiene las clases agregadas.



Agregación en UML.

- UML permite representar una relación entre una clase y sus elementos agregados, esto quiere decir que las clases agregadas no afectan el funcionamiento directo de la clase que las contiene.
- El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

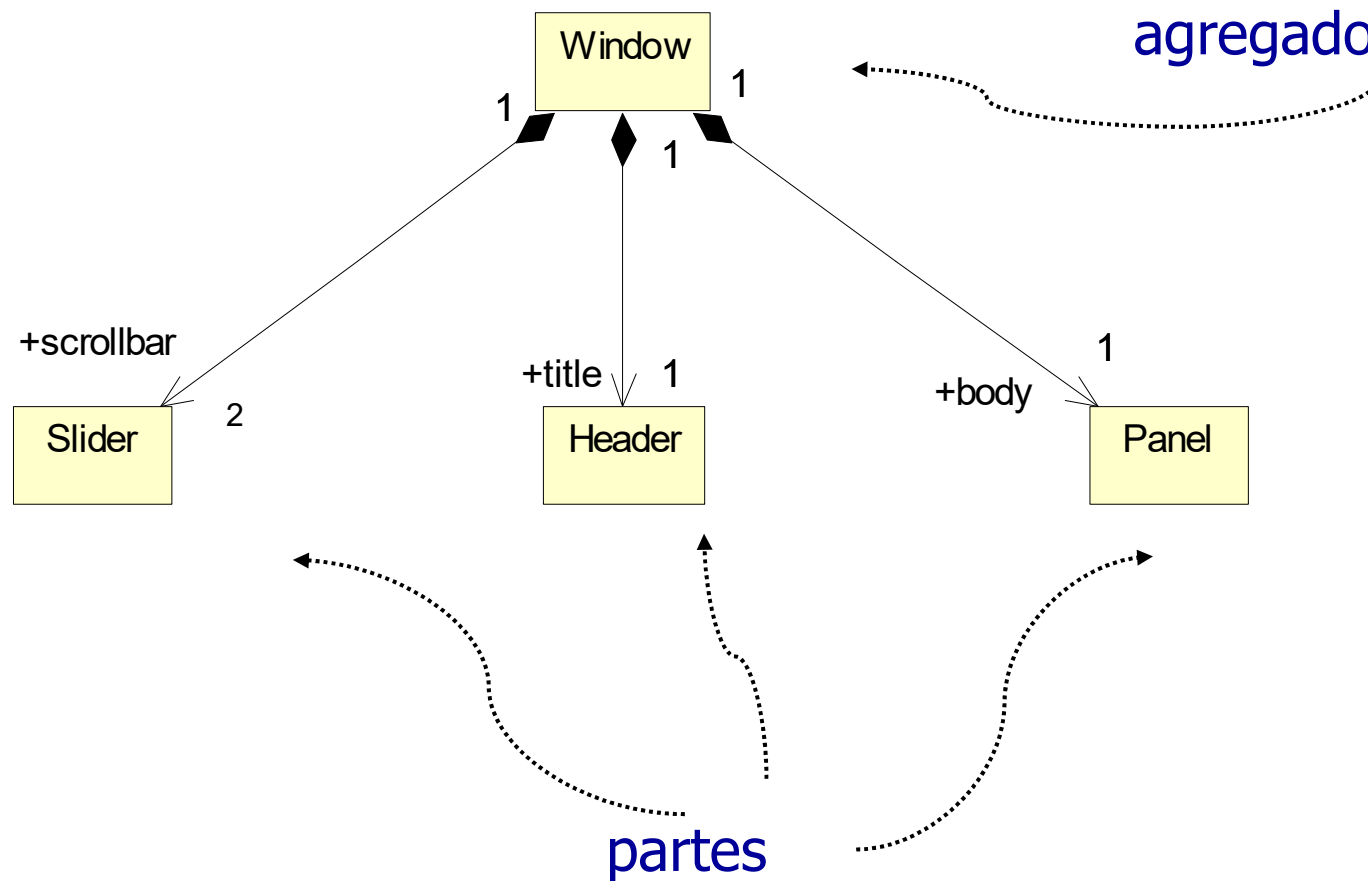


- Relación estructural *parte-de*

Relaciones entre clases. Composición

- Forma “fuerte” de agregación con ciclo de vida coincidente.
- A diferencia de la agregación representa una clase que está compuesta por otras clases que son indispensables para que esta exista o funcione
- Una parte pertenece a un único agregado (**exclusividad**)
- Si se elimina un agregado se eliminan todas sus partes (**dependiente**)
- Una parte se puede añadir o eliminar en cualquier instante al agregado.
- El agregado gestiona la creación y destrucción de las partes

Composición en UML.



Ejemplo Agregación y Composición

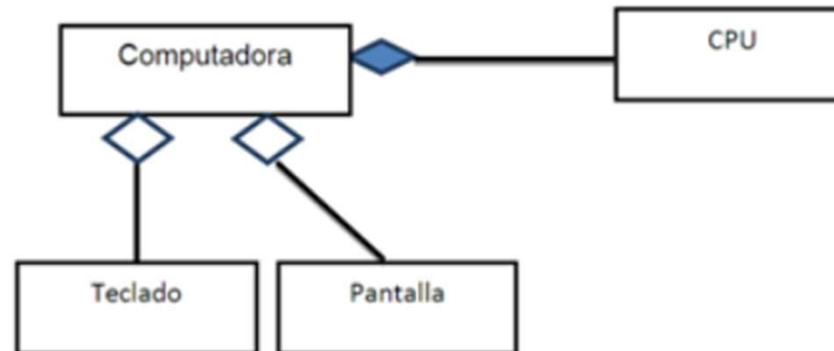
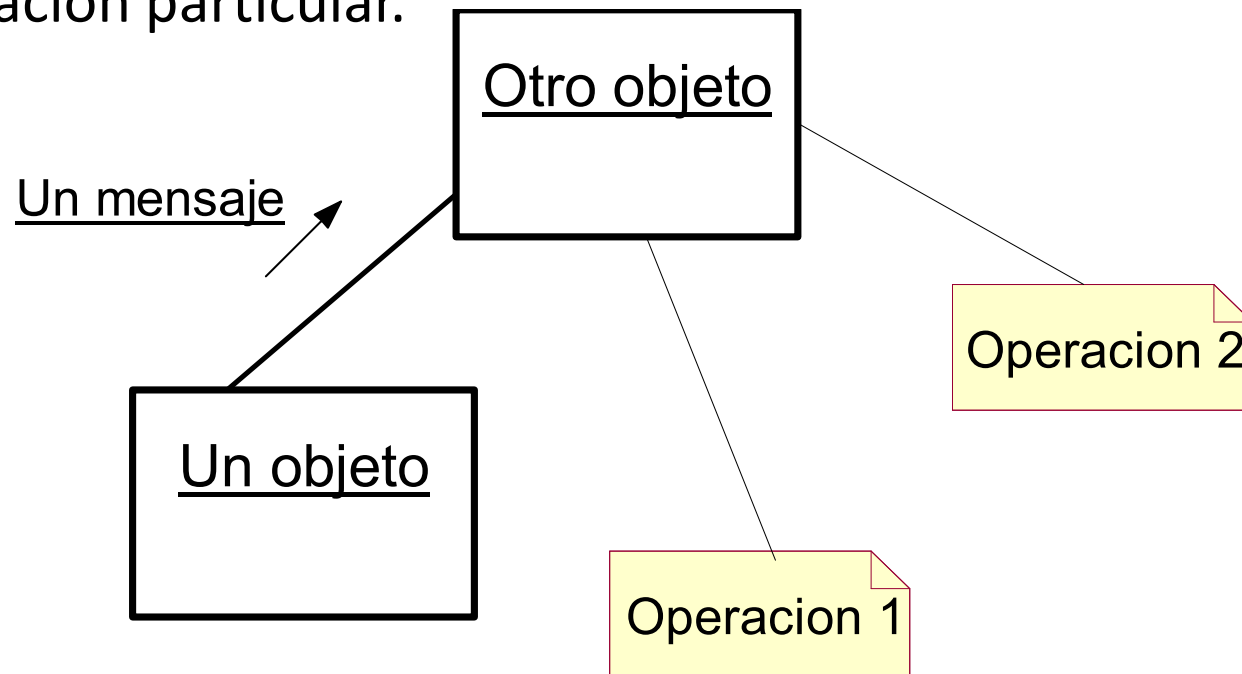


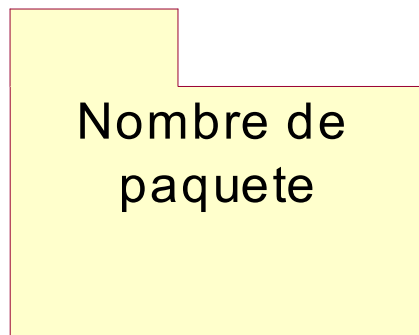
Diagrama de Objetos en UML

- En un Diagrama de clases, las Clases definen qué objetos existirán durante la ejecución del Sistema.
- Un **diagrama de objetos UML** muestra un conjunto de objetos en un determinado instante de la ejecución.
- Mientras el diagrama de clases define todas las posibles configuraciones del sistema, un diagrama de objetos define una configuración particular.



Packages (o Paquetes) en UML

- Los paquetes ofrecen un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado



Packages en UML

- Cada paquete corresponde a un submodelo (subsistema) del modelo (sistema)
- Un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a (está definido en) sólo un paquete
- Una clase de un paquete puede aparecer en otro paquete por la importación a través de una **relación de dependencia** entre paquetes

Packages en UML

- Todas las clases no son necesariamente visibles desde el exterior del paquete, es decir, un paquete encapsula a la vez que agrupa

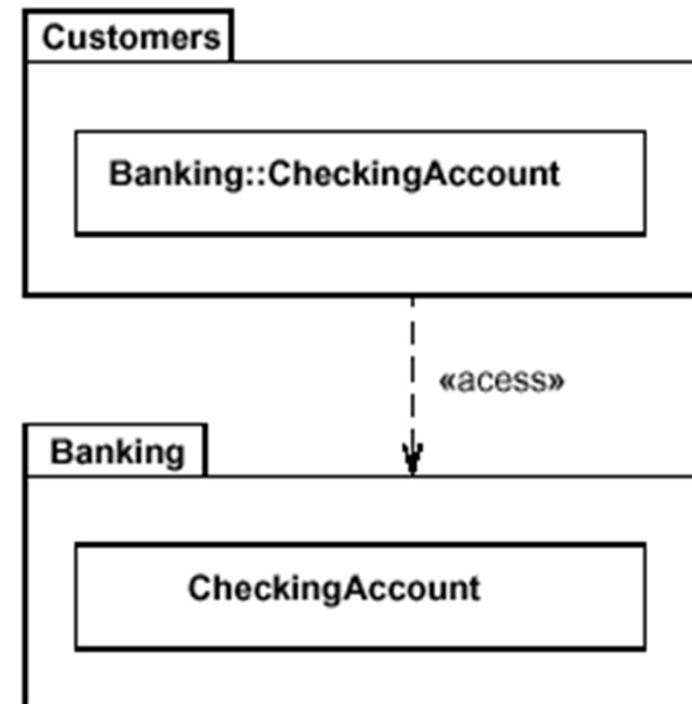
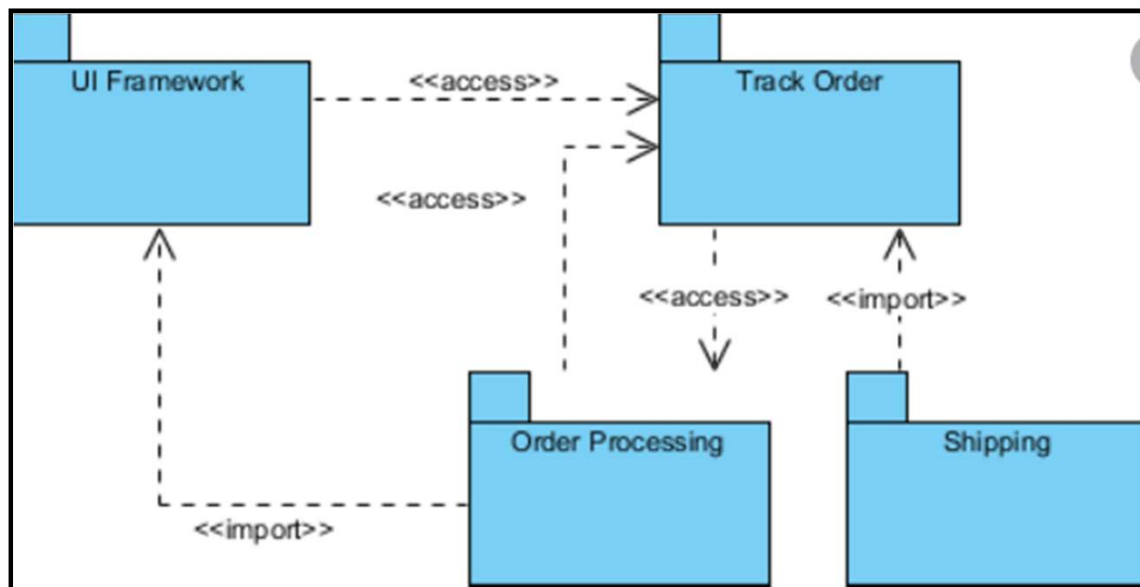


Diagrama de Packages en UML



Bibliografía

- Wirfs R., Brock “Designing Object-Oriented Software”.