



El futuro digital
es de todos

MinTIC



Vigilada Mineducación



CICLO III: Desarrollo de Software

Hechos
QUE CONECTAN 

Sesión 02:

Desarrollo de Software

**Modelado y Desarrollo de software y conceptos
básicos de modelo de comunicaciones TCP/IP**

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Diseñar un sistema de software basado en una metodología de desarrollo a partir de requerimientos funcionales de un tercero.
2. Explicar el modelo de comunicaciones TCP/IP.
3. Exponer la relevancia del modelo de comunicaciones TCP/IP para el desarrollo de aplicaciones basadas en el modelo cliente y servidor.
4. Describir el sistema de nombres de dominio (DNS).
5. Describir otros protocolos de comunicación como TCP, UDP, IP.

Modelado y Desarrollo de Software

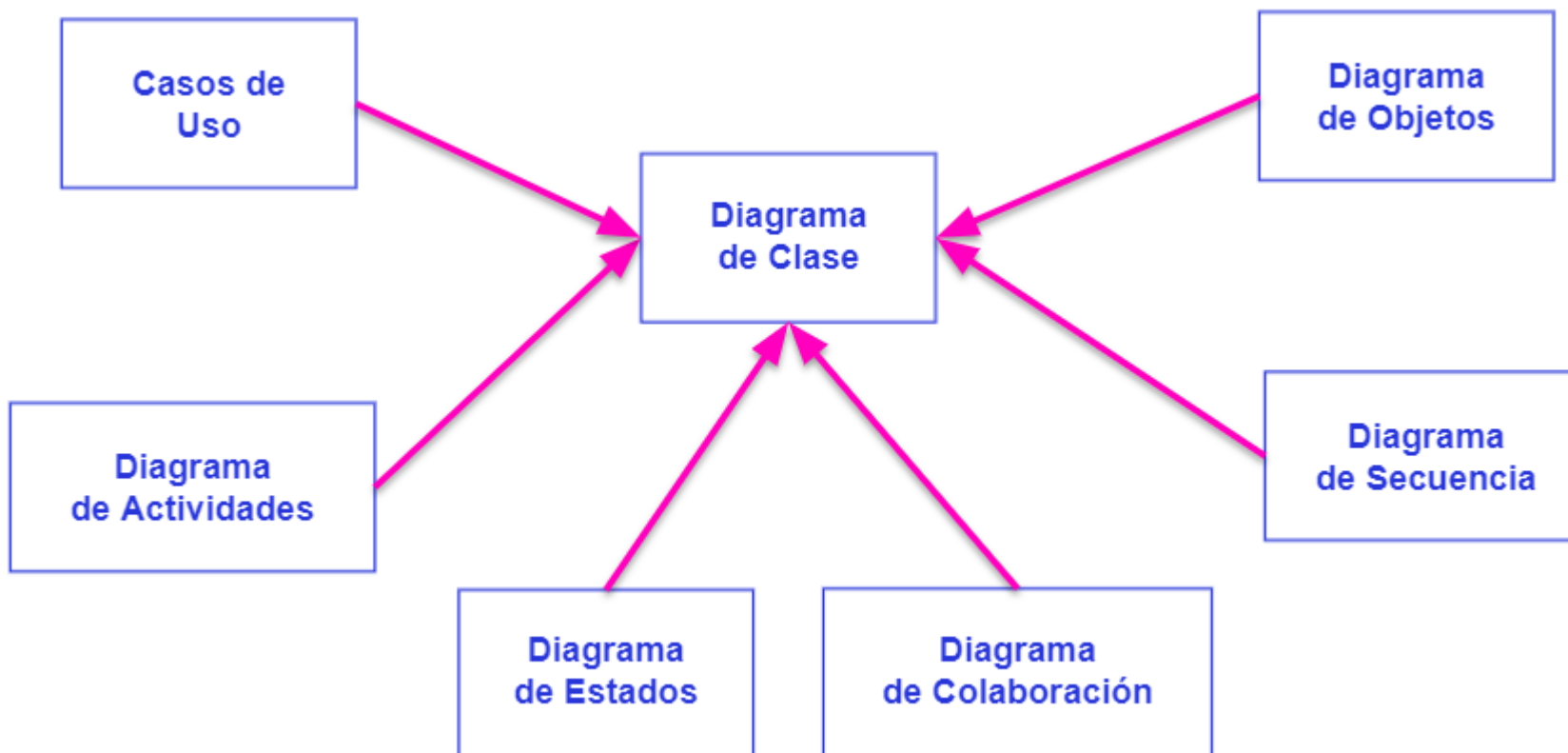
Diagrama de clases

- Los diagramas de clase muestran la vista estática de un conjunto de clases, interfaces, colaboraciones y sus relaciones.
- Son los diagramas más comunes y por tanto más utilizados en lo que respecta al modelado de sistemas orientados a objetos.

Partes del diagrama de clases

- **Atributos:** Características de una clase.
- **Operaciones:** Comportamiento de una clase.
- **Asociación:** Relación entre clases.
- **Herencia:** Organización de las definiciones de la clase para simplificar y facilitar su implementación.

Todos los diagramas soportan el diagrama de clases



Relaciones entre clases

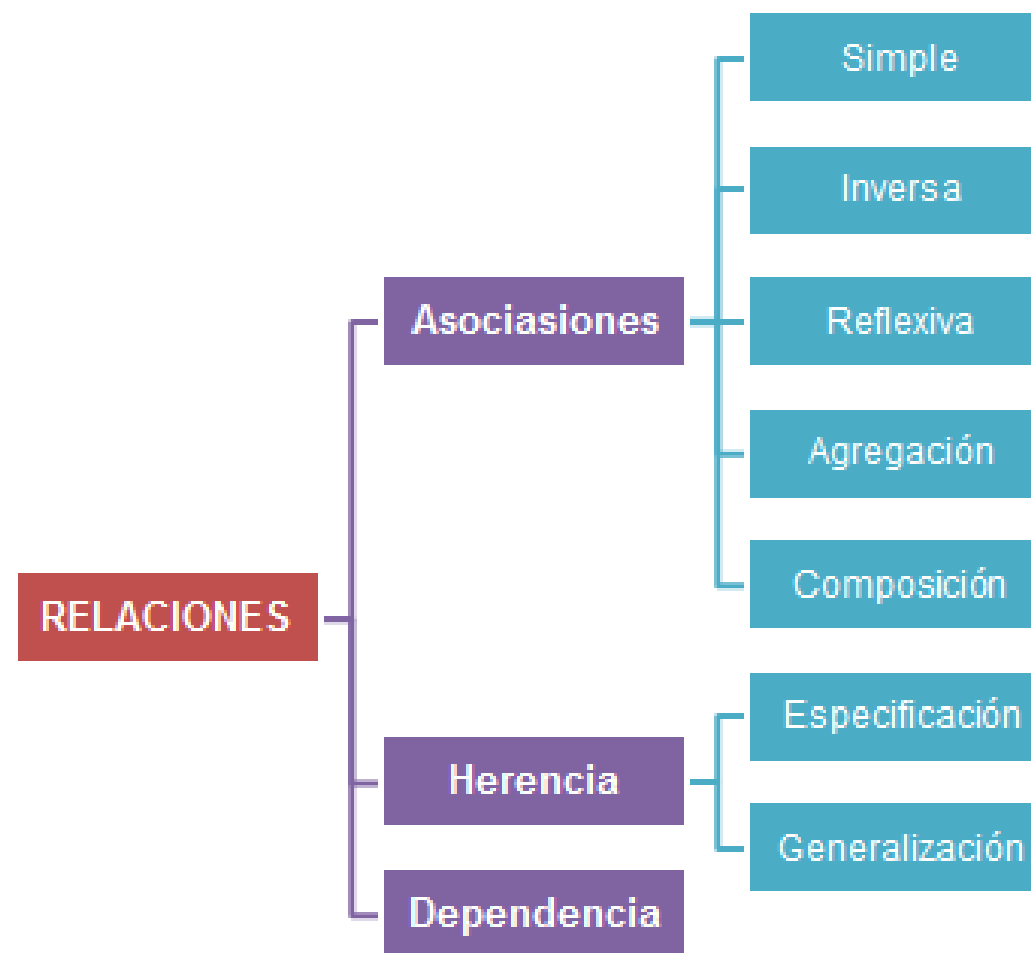


Diagrama de clases - Asociaciones

- El objetivo de la asociación es mostrar cómo los objetos de una clase se relacionan con objetos de otra clase. Por ejemplo:
 - Una persona tiene un perro
 - Una persona cuida un perro
- Multiplicidad: Detalla cuántos objetos van a participar en la relación
- La dirección en las flechas de la asociación determinan en qué dirección puede recorrer una asociación en el momento de la ejecución.
- Una asociación sin flechas significa que se puede ir de un objeto a otro y viceversa.

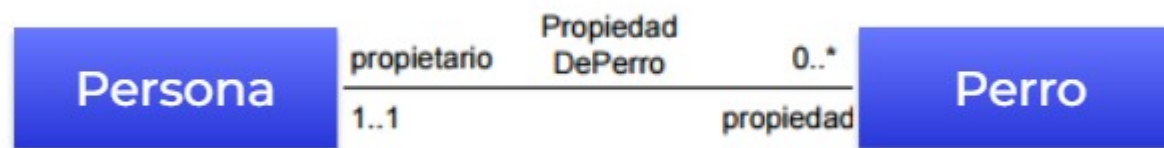


Diagrama de clases - Asociación reflexiva

- Una clase puede asociarse con sí misma.
- Lo que significa que una instancia de la clase está relacionada con otra instancia de la misma clase, más no, que la instancia está relacionada consigo misma.
- Ejemplo:
 - Clase Empleado puede relacionarse con sí misma a través del rol gerente/dirige.

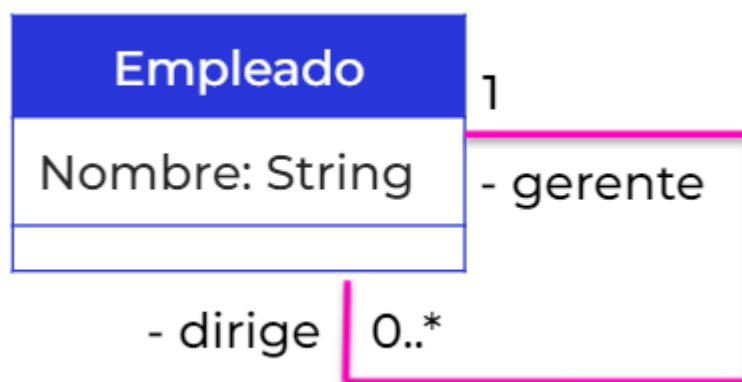


Diagrama de clases - Agregación y composición

- Cada agregación es un tipo de asociación.
- Cada composición es una forma de agregación.

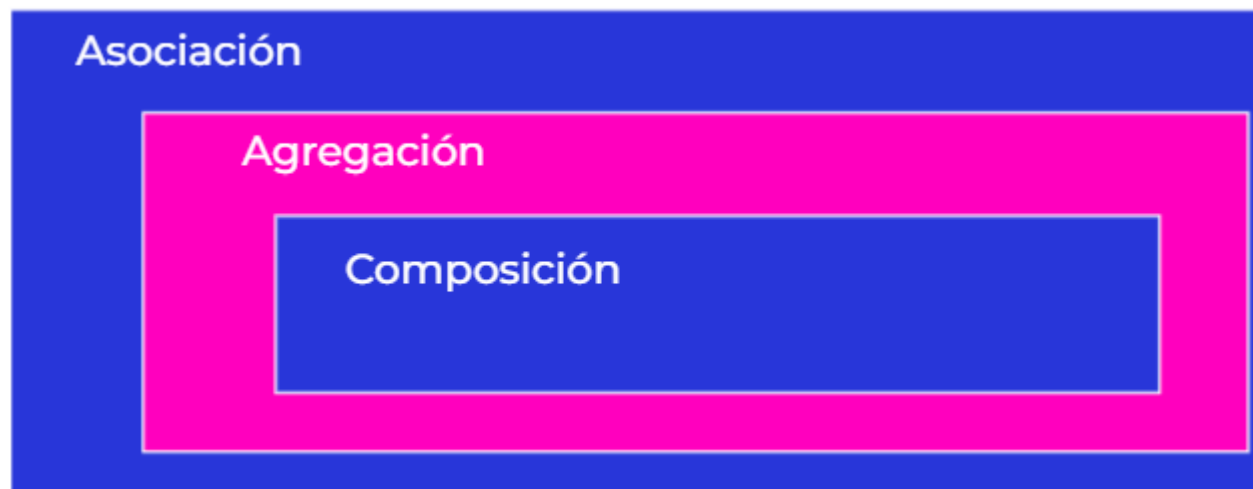


Diagrama de clases - Agregación

- Es un tipo de asociación que indica que una clase es parte de otra clase (de un todo). Es decir, los componentes pueden ser compartidos por varios compuestos.
- Como se puede observar en el ejemplo, Empresa es una entidad “whole/todo” y Cliente es una parte de la Empresa.
- La agregación se representa en UML mediante un diamante de color blanco colocado en el extremo en el que está la clase que representa el “todo”.
- En esta asociación, la clase agregada puede permanecer sin la clase ‘todo’ o principal.

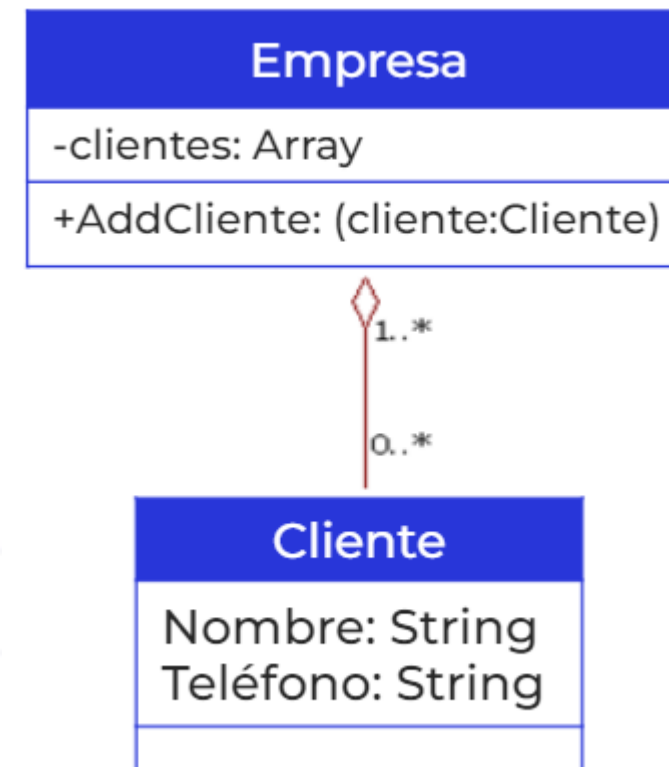


Diagrama de clases - Composición

- La composición implica que los componentes de un objeto sólo pueden pertenecer a un solo objeto agregado, de forma que cuando el objeto agregado es eliminado todas sus partes son eliminadas también.
- Para representarla el diamante debe estar relleno.
- En el ejemplo, un objeto Empresa está compuesto por uno o varios objetos del tipo empleado y el tiempo de vida de los objetos empleados depende del tiempo de vida de Empresa, ya que si no existe una Empresa no pueden existir sus empleados.
- Otra característica a tener en cuenta es que la clase contenida sólo puede relacionarse con una instancia de la clase contenedora.

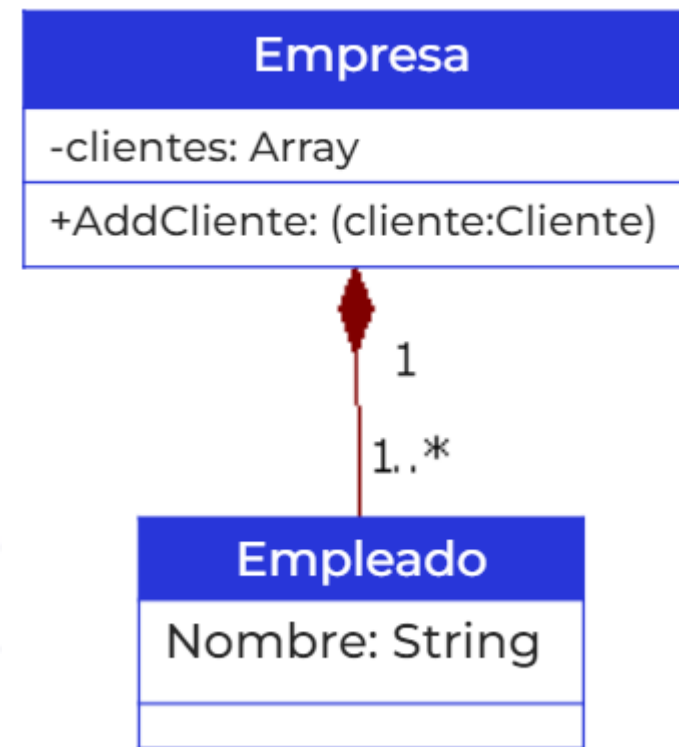


Diagrama de clases - Herencia

- La herencia es la ya conocida relación de generalización y especificación entre clases.
- En UML la herencia se representa mediante una flecha, cuya punta es un triángulo vacío.
- La flecha que representa a la herencia va orientada desde la subclase a la superclase.
- No hay multiplicidad ni roles.

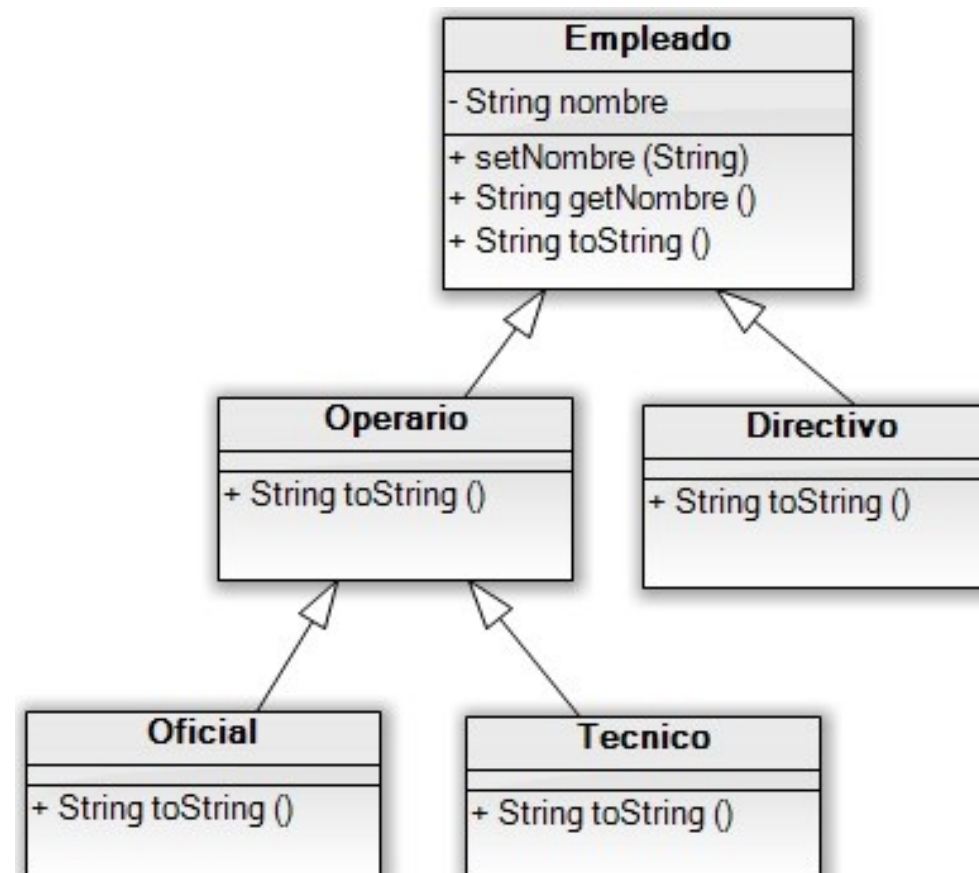


Diagrama de clases - Dependencia

- Una dependencia se puede definir como una relación semántica entre dos clases en la cual un cambio de un elemento (independiente) puede afectar la semántica de otro (dependiente).
- Representación gráfica:

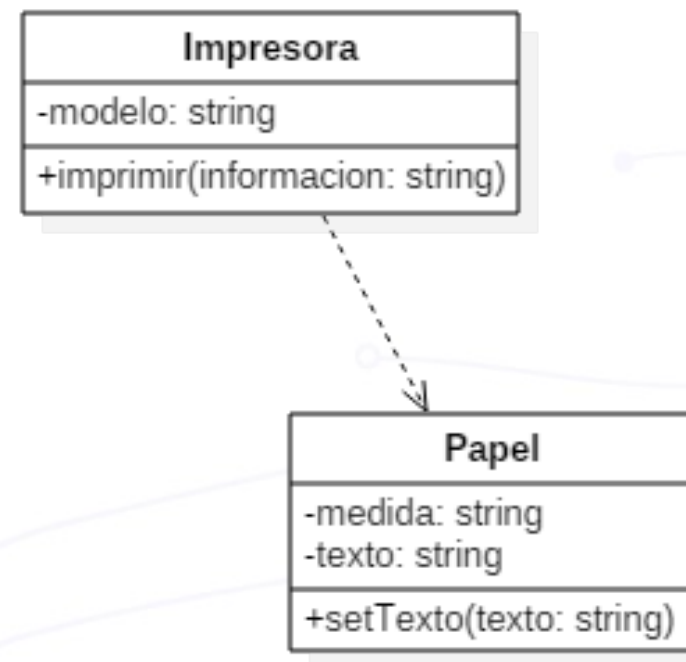
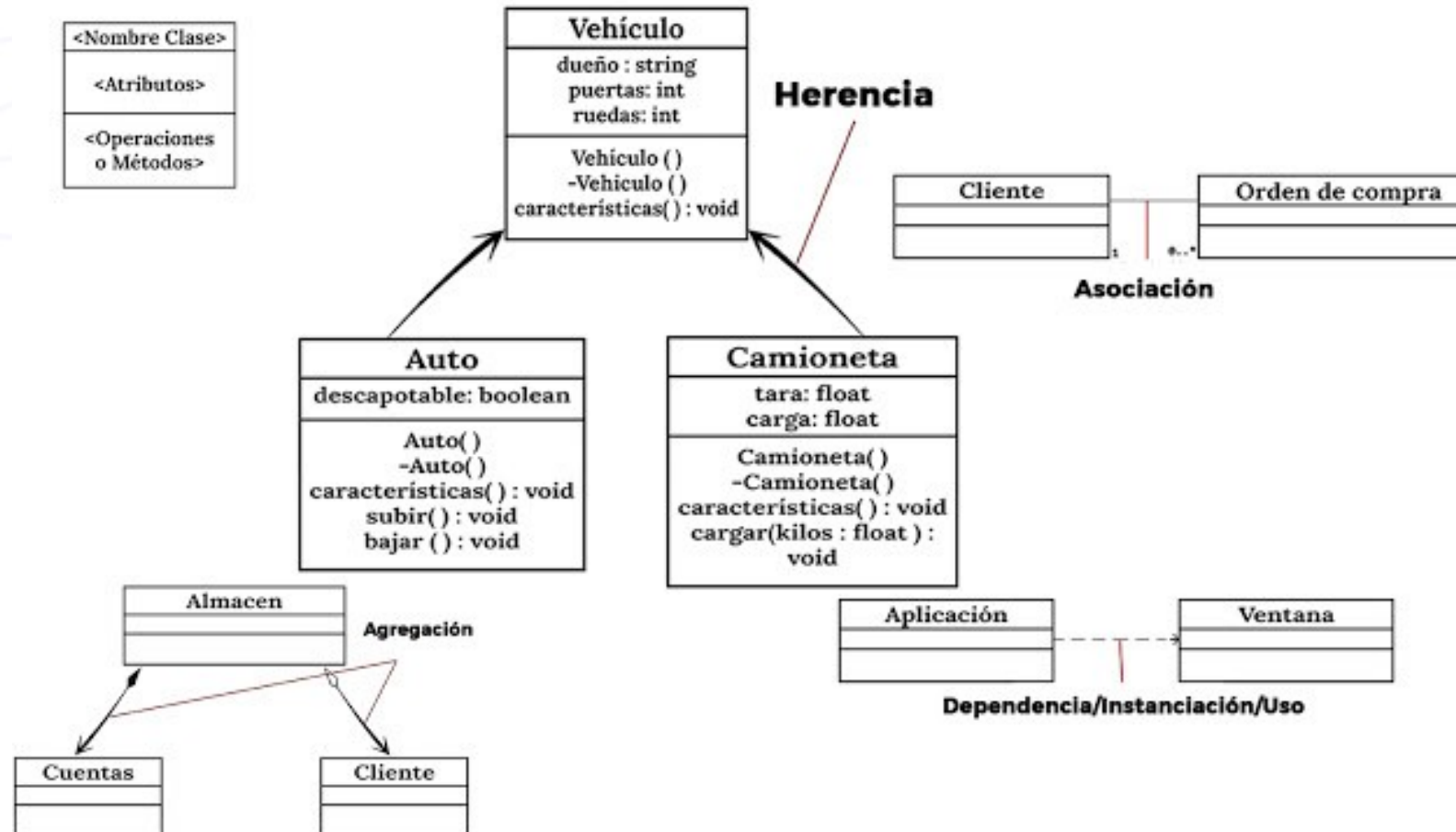




Diagrama de clases - Ejemplo

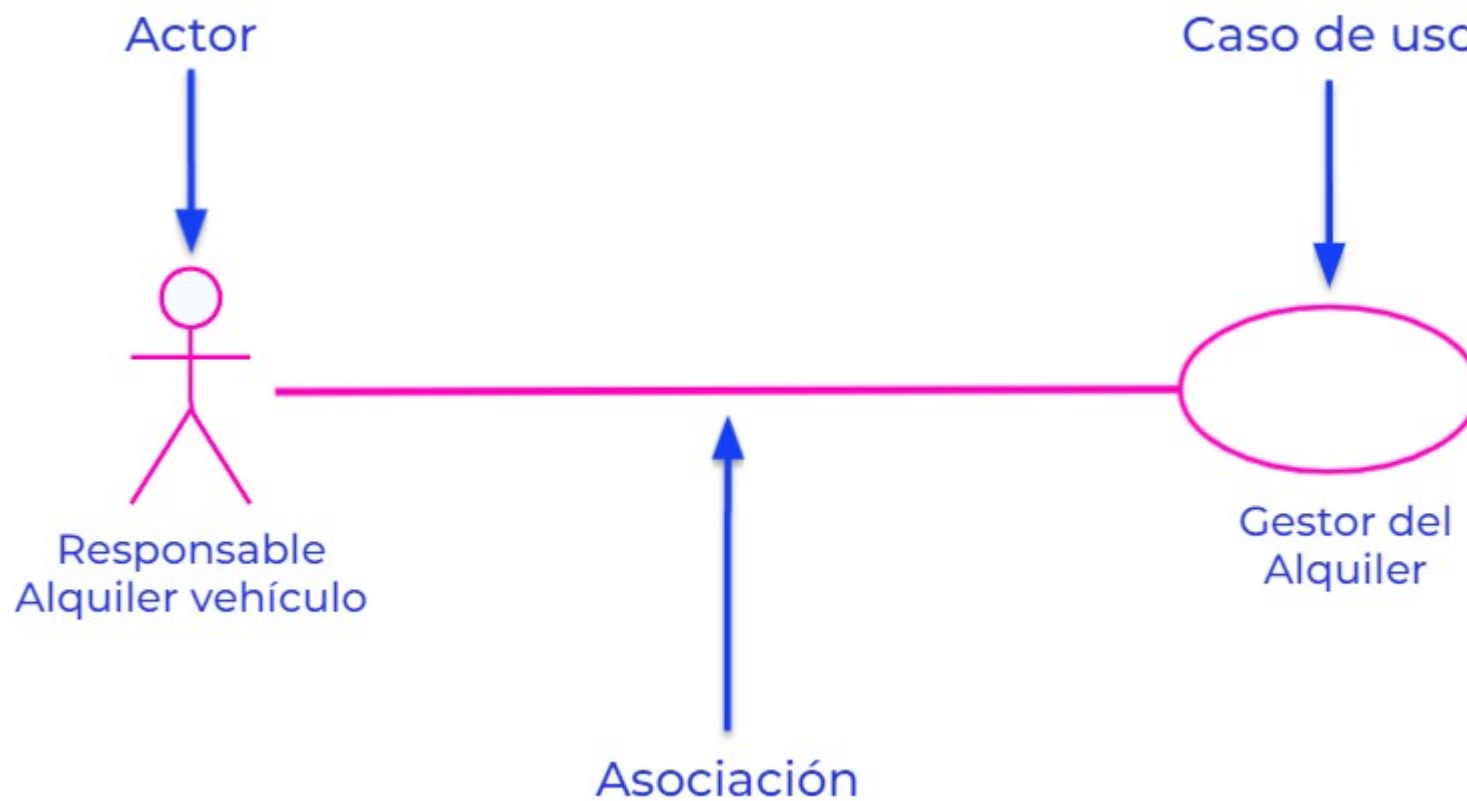


- **Nota:** Recuerde que la composición es un tipo de agregación.

Modelado de casos de uso

- Un caso de uso detalla un comportamiento esperado del sistema.
- Representan los requisitos funcionales del sistema.
- “Un caso de uso especifica un conjunto de secuencias de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un particular actor.” (Definición en UML).
- Describen qué hace el sistema, no cómo lo hace.
- Partes de un caso de uso (cdu):
 - Conjunto de secuencias de acciones.
 - Actores, roles que pueden jugar los usuarios.
 - Variantes: versiones especializadas, un cdu que extiende a otro o un cdu que incluye a otro.

Ejemplo de caso de uso



Actores

- Representan un conjunto de roles que desempeñan los usuarios de los casos de uso al interaccionar con el sistema.
- Roles representados por personas, dispositivos, u otros sistemas.
- Un actor puede ser el tiempo (“procesos iniciados automáticamente por el sistema”).
- No forman parte del sistema.
- Pueden representar diferentes roles.
- En la elaboración de un caso de uso pueden interactuar distintos actores.
- Pueden estar en diferentes casos de uso.
- Necesitan el caso de uso y participar en este.

Actores - Tipos

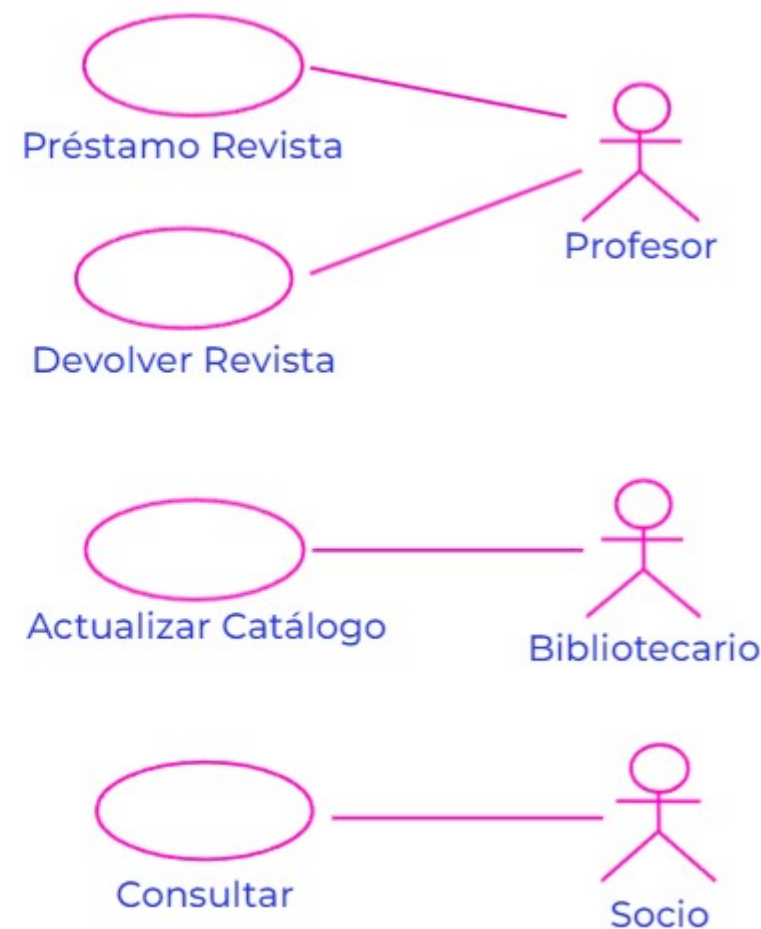
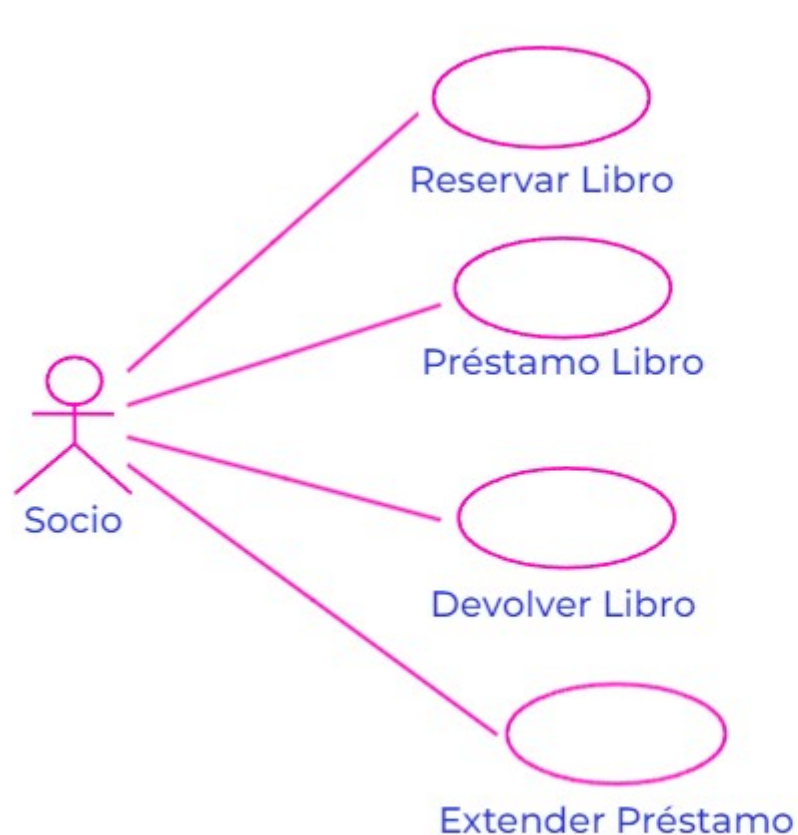
Dos tipos de actores:

- **Principal:**
 - Le solicita al sistema el cumplimiento de un objetivo.
- **Secundarios:**
 - El sistema los necesita para satisfacer un objetivo.

Propiedades de los casos de uso

- Son iniciados por un actor con un objetivo en mente y es completado con éxito cuando el sistema lo satisface.
- Puede incluir secuencias alternativas que llevan al éxito y fracaso en la consecución del objetivo.
- El sistema es considerado como una “caja negra” y las interacciones se perciben desde fuera.
- El conjunto completo de casos de uso especifica todas las posibles formas de usar el sistema, esto es el comportamiento requerido.

Diagrama de casos de uso - Ejemplo



Relaciones de casos de uso

Tres tipos de relaciones:

- **Generalización:**

- Un cdu hereda el comportamiento y significado de otro.

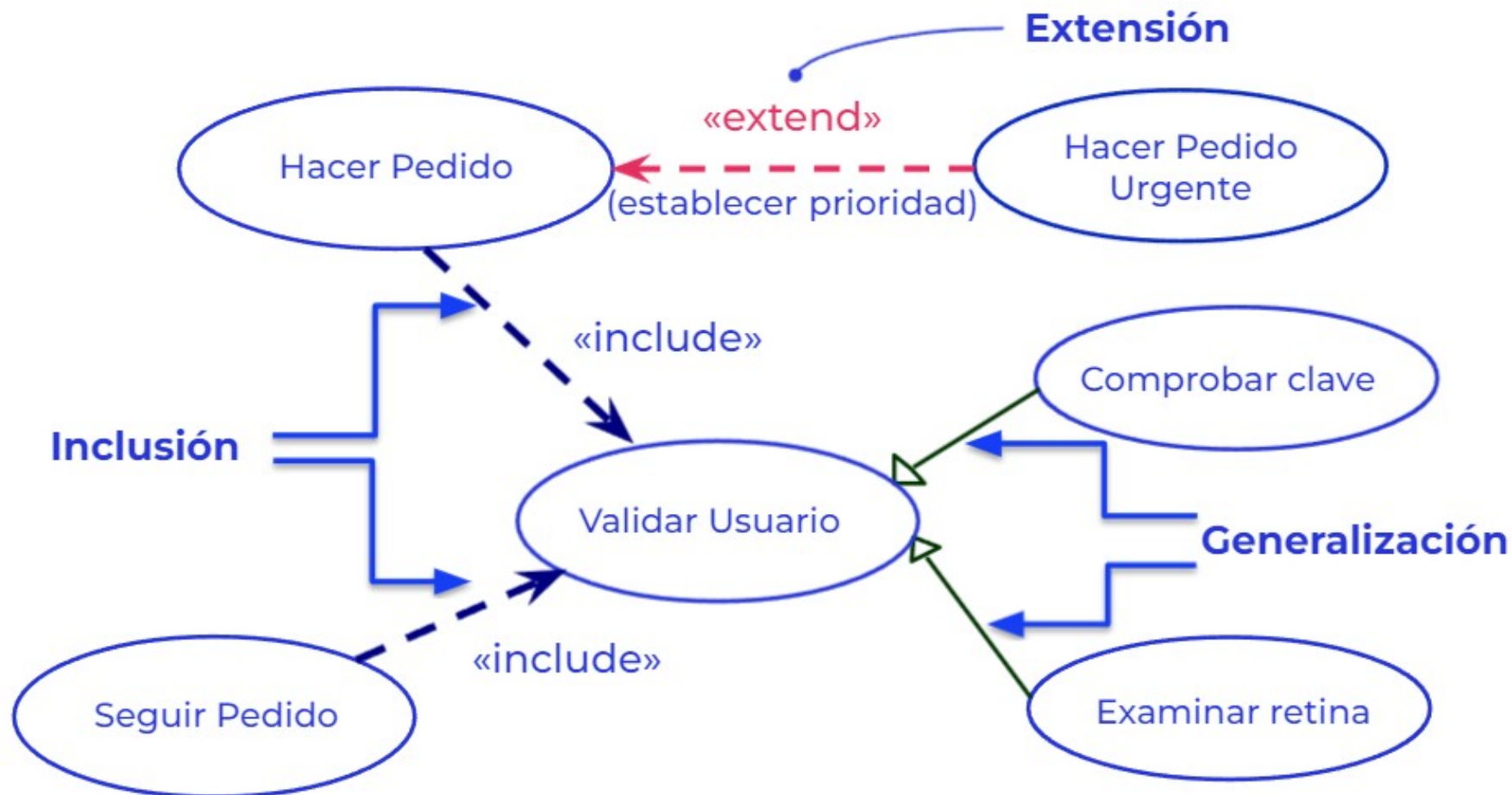
- **Inclusión:**

- Un cdu base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.

- **Extensión:**

- Un cdu base incorpora implícitamente el comportamiento de otro cdu en el lugar especificado indirectamente por este otro cdu.

Relaciones de casos de uso - Ejemplo



Construcción de casos de uso - Pasos

1. Identificar los usuarios del sistema.
2. Encontrar todos los roles que representan los usuarios y que son importantes al sistema.
3. Identificar para cada rol todas las formas de interactuar con el sistema.
4. Por cada objetivo crear un caso de uso.
5. Estructurar los casos de uso.
6. Revisar y validar con el usuario.

Conceptos básicos de modelo de comunicaciones TCP/IP

Modelo OSI

- Open Systems Interconnection.
- Se define como una arquitectura necesaria para establecer comunicación entre computadoras.
- Es un modelo de referencia que define cómo se debe transmitir un mensaje entre nodos en una red de datos.
- Cuenta con 7 capas de funciones.
- Su objetivo es estandarizar el proceso de comunicación entre dos usuarios en una red, dividido por capas.

Modelo TCP/IP

- Es un conjunto de protocolos que permite que los computadores se comuniquen entre sí.
- El nombre se deriva de dos de los protocolos que lo conforman:
 - Transmission Control Protocol (TCP).
 - Internet Protocol (IP).
- TCP/IP son los protocolos fundamentales de Internet.

Arquitectura de TCP/IP

Aplicación	Aplicaciones y procesos que usa la red.
Transporte	Servicios de entrega de datos entre nodos.
Internet	Define el datagrama y maneja el enrutamiento.
Acceso de Red	Procesos para acceder al medio físico.

Modelo TCP/IP



- **Capa de Aplicación:**

- Contiene las funciones de las capas de Aplicación, Presentación y Sesión del Modelo OSI.
- Representa los Datos.
- Encriptación.
- Control de Diálogo.

- **Protocolos:**

- FTP - File Transfer Protocol (TCP).
- HTTP - Hypertext Transfer Protocol (TCP).
- SMTP - Simple Mail Transfer Protocol (TCP).
- DNS - Domain Name Service (UDP).
- TFTP - Trivial File Transfer Protocol (UDP).

Modelo TCP/IP



- **Capa de Transporte:**
 - Responsable de la calidad de servicio (TCP):
 - Confiabilidad:
 - Recuperación de Fallas.
 - Acknowledgment.
 - Control de Flujo:
 - Sliding Windows.
 - Orientado a Conexión.
- **Protocolos:**
 - TCP - Transmission Control Protocol.
 - UDP - User Datagram Protocol (No orientado a la conexión).

Modelo TCP/IP



- **Capa de Internet:**
 - Usa el protocolo IP.
 - Se determina la Ruta.
 - Enrutamiento de Paquetes.
 - Direccionamiento.
 - No orientado a conexión (Best-effort delivery protocol).
- **Protocolos:**
 - IP - Internet Protocol.

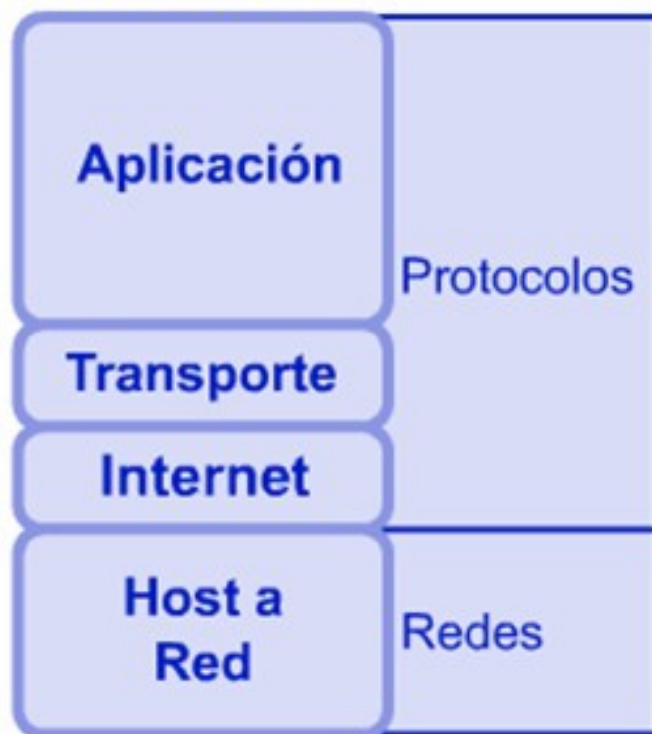
Modelo TCP/IP



- **Capa de Red (Host a Red):**
 - Contiene las funciones de la capa de Enlace de Datos y física del Modelo OSI.
 - Tecnologías WANs y LANs como Frame Relay y Ethernet.
 - Encargada de todos los procesos requeridos para realizar el enlace físico.
- **Tecnologías:**
 - LAN & WAN Technologies.
 - Ethernet.
 - WI-FI IEEE 802.11.

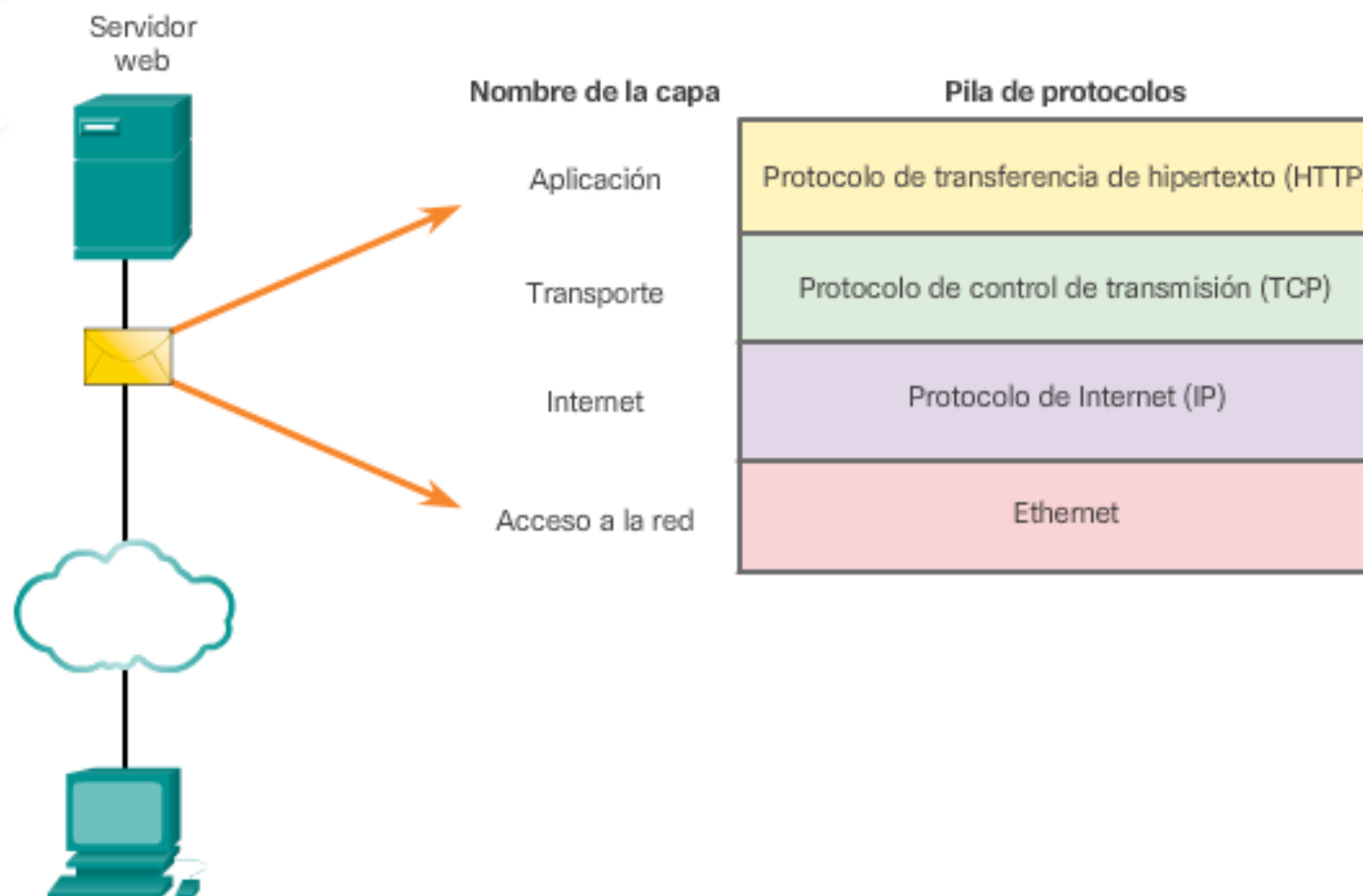
Modelo OSI vs TCP/IP

- Modelo TCP/IP



- Modelo OSI

Esquema Cliente-Servidor



Esquema Cliente-Servidor

No existe una definición única de este modelo, se puede definir de varias formas:

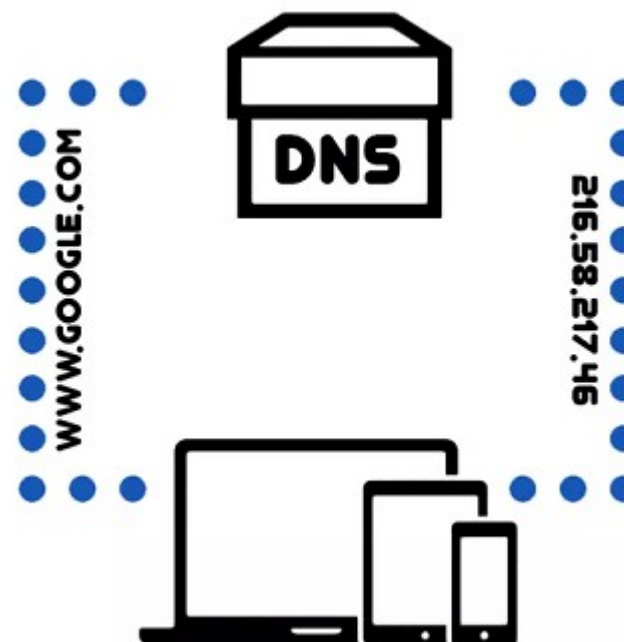
- Como una técnica para la distribución de recursos entre computadoras personales.
- Como una tecnología distribuida que determina el papel de un cliente que requiere el servicio y de un servidor que se lo proporciona.
- Como un conjunto de funciones:
 - Los servidores ofrecen una interfaz pública y bien definida a todos los clientes.
 - Las aplicaciones de los clientes no conocen la lógica del servidor, solamente su interfaz.
 - Las aplicaciones de los clientes deben ser independientes de su localización.
 - Los clientes no dependen de la ubicación del servidor.

Esquema Cliente-Servidor

- Los procesos que utilizan protocolos TCP/IP se efectúan según el esquema Cliente-Servidor.
- Los clientes y servidores son procesos que pueden estar en el mismo o distinto computador.
- Los procesos cliente utilizan los servicios proporcionados por los servidores.
- Tanto los elementos de infraestructura como los de red deben incorporarse en los clientes y en los servidores.

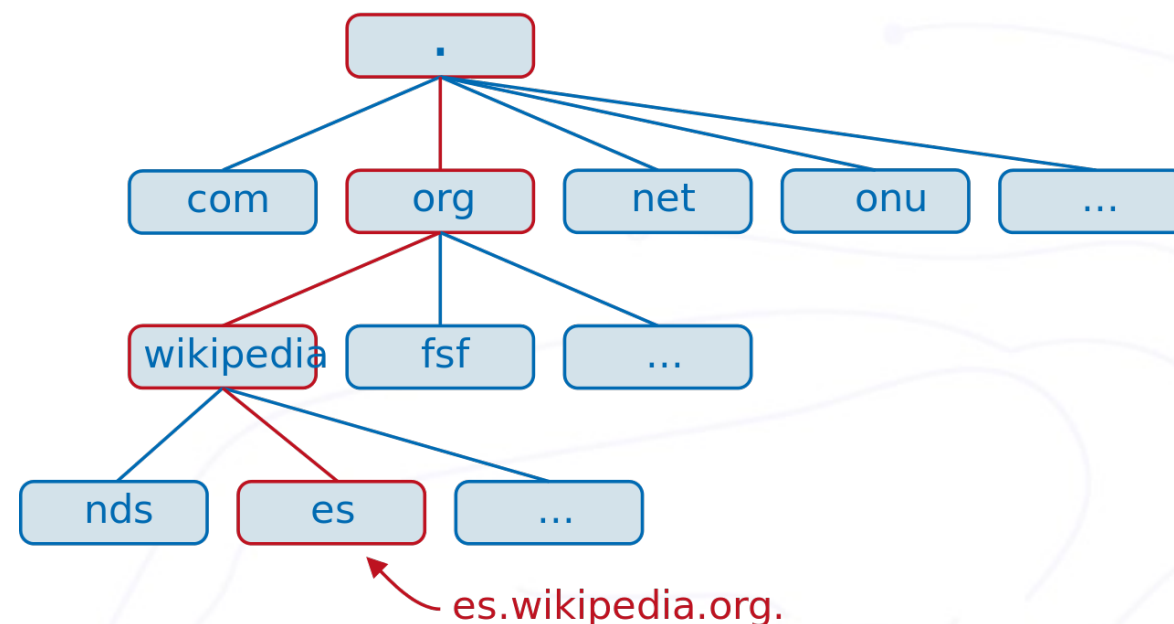
Sistema de Nombres de Dominio (DNS)

- El sistema de nombres de dominio (DNS por sus siglas en inglés) es un servicio de nombres que convierte el dominio de usuario conocido (por ejemplo, "www.paginaprincipal.com") a la dirección IP correspondiente. Al enviar el nombre de dominio como una consulta en la barra de direcciones, el dominio se traduce desde el DNS para luego transferir la solicitud al servidor correspondiente.
- Un dominio está conformado por:
 - Raíz DNS (punto).
 - Dominio de primer nivel (TDL).
 - Dominio de segundo nivel (SDL).
 - Uno o más subdominios.
 - Nombre de host (hostname).



Sistema de Nombres de Dominio (DNS) - Jerarquía

- El espacio de nombres de dominio tiene una estructura tipo árbol.
- Las hojas y los nodos del árbol representan las etiquetas de los medios.
- Un nombre de dominio completo consiste en la concatenación de todas las etiquetas de un camino.
- Un nombre de dominio debe incluir todos los puntos y tiene una longitud máxima de 255 caracteres.
- Un nombre de dominio se escribe siempre de derecha a izquierda.



Protocolos TCP, UDP e IP

Protocolo IP

- El protocolo IP se encarga de:
 - **Direcciones IP:** Identificar equipos en la conexión extremo a extremo. Actualmente se emplean direcciones de versión 4 (32 bits) y versión 6 (128 bits).
 - **Enrutamiento:** Definir el camino que debe utilizar un paquete, basándose en la dirección IP del destinatario.
 - **Formato de paquetes:** Agrupar paquetes en unidades conocidas como datagramas.
 - **Fragmentación:** En caso que un paquete sea demasiado grande para su transmisión a través del medio de red, el protocolo IP del sistema de envío divide el paquete en fragmentos de menor tamaño.

Protocolos TCP, UDP e IP

Protocolo TCP

- Permite que las aplicaciones se comuniquen entre sí como si estuvieran conectadas físicamente.
- Envía los datos en un formato que se transmite carácter por carácter, en lugar de transmitirse por paquetes. Cuando todo ha sido enviado exitosamente se procede al cierre de la conexión.
- Es necesario para la descarga Web o la transferencia de ficheros.

Protocolos TCP, UDP e IP

Protocolo UDP

- Proporciona un servicio de entrega de datagramas.
- No verifica las conexiones entre los hosts transmisores y receptores.
- Dado que el protocolo UDP elimina los procesos de establecimiento y verificación de las conexiones, resulta ideal para las aplicaciones que envían pequeñas cantidades de datos o cuando el volumen de datos es elevado y éste tiene que llegar en tiempo real.
- VoIP es un ejemplo de aplicación que utilizará UDP.

Ejercicios de Práctica



El futuro digital
es de todos

MinTIC

Misión
TIC 2022

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS

**POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!**

Hechos

QUE

CONECTAN ✓

