



**TECNOLÓGICO
DE MONTERREY®**

TC2026 – Desarrollo de aplicaciones Web

MCC Fernanda Montiel Lerma

fernanda.montiel@itesm.mx

fernanda.montiel@voxelout.com.mx

Modelo C-S

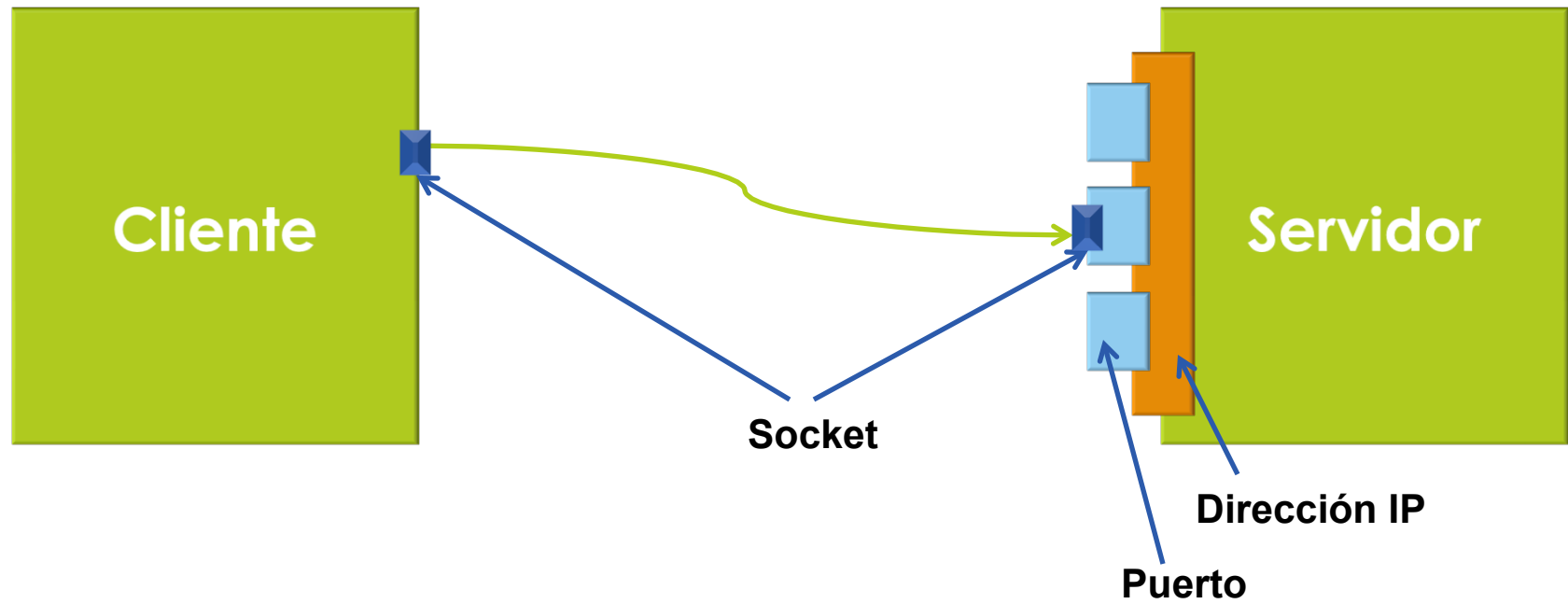
Aplicaciones C/S

■ Servidor

- Escucha peticiones.
- Es localizado por una dirección y un puerto.
- Entiende un protocolo.
- Responde a peticiones del servicio:
 - Proporciona un recurso Web o envía un mensaje de correo electrónico a otro punto.

■ Cliente

- Necesita conocer la dirección, el puerto y el protocolo.
 - Análoga: Conocer el num de teléfono, la extensión y el idioma.
- Solicita el servicio y recibe respuestas usando el protocolo requerido por el servidor.



Aplicaciones C/S

■ Socket

- A nivel de transporte, mantiene la entrega de paquetes TCP entre emisor y receptor. (UDP no usa sockets porque es un protocolo sin garantía de entreg).
- Análogo a una puerta: El proceso emisor envía un mensaje por esa puerta.
 - Asume la existencia de una infraestructura de transporte (TCP) que llevará el mensaje al otro extremo, al socket del receptor.

Aplicaciones C/S

- Implementación de C/S en Java

Distribuciones

- CS
 - Cliente inicia la petición.
- Push
 - Servidor notifica a clientes suscritos.
- Peer to Peer
 - El host actúa tanto como cliente como servidor del mismo servicio.

Protocolos

- ▣ Idioma entre las aplicaciones o servicios
- ▣ Define qué y cómo debe hacerse una solicitud, así como el tipo de respuesta que se puede esperar (incluyendo errores).
- ▣ HTTP es un protocolo CS
 - ▣ Tu Web App NO se considera una aplicación cliente/servidor, pero se ejecuta a través de uno (todos los navegadores hablan HTTP).



HTTP



HTTP

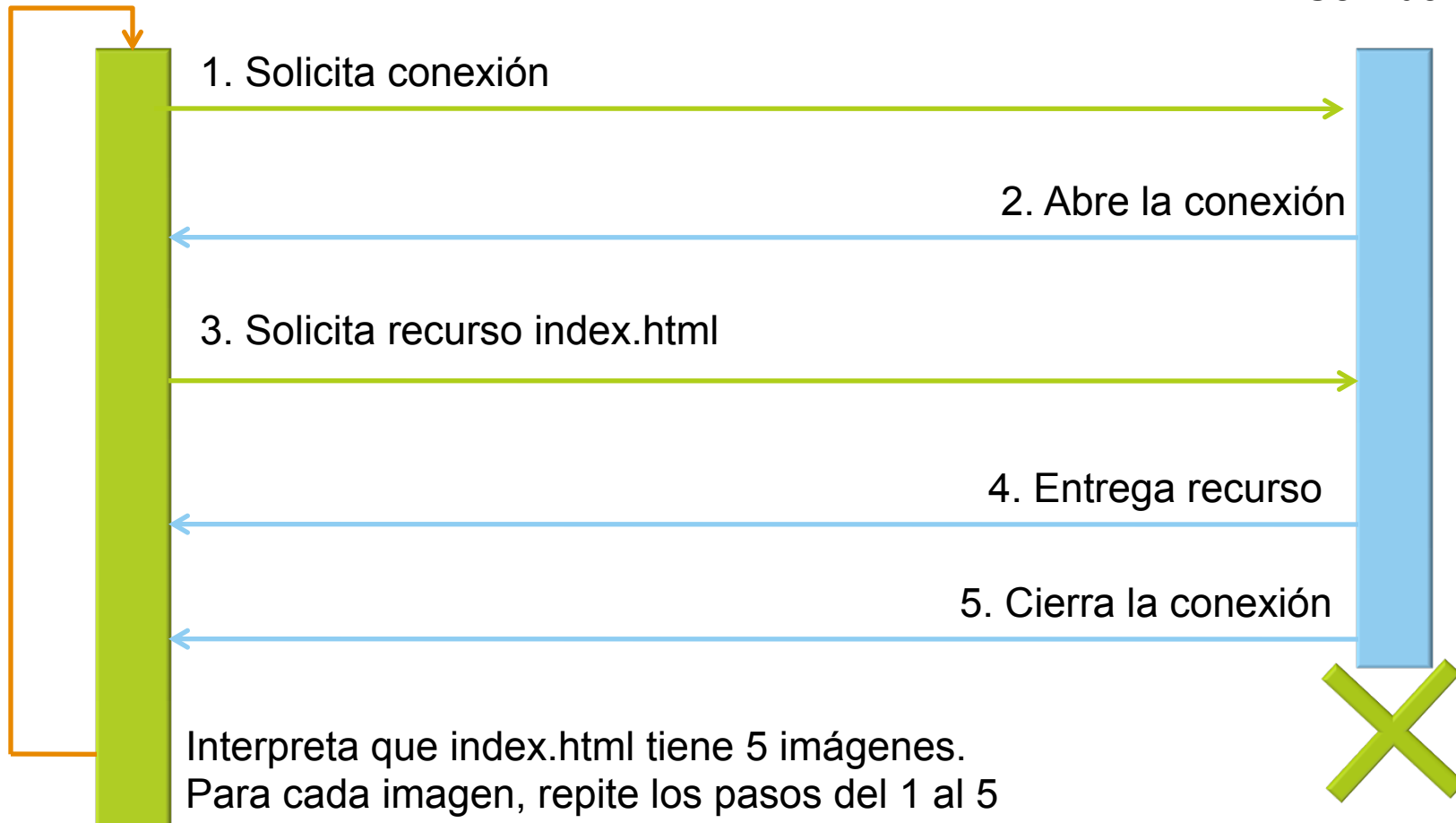
1. El navegador (cliente) inicia la petición a una dirección y un puerto, vía TCP.
 - ▣ La dirección puede estar enmascarada en una URL, registrada en tablas DNS.
2. El servidor acepta la conexión.
3. El cliente envía la petición HTTP de algún recurso.
4. El servidor responde con un objeto.
5. La conexión se cierra.

HTTP

Navegador

HTTP no persistente

Servidor

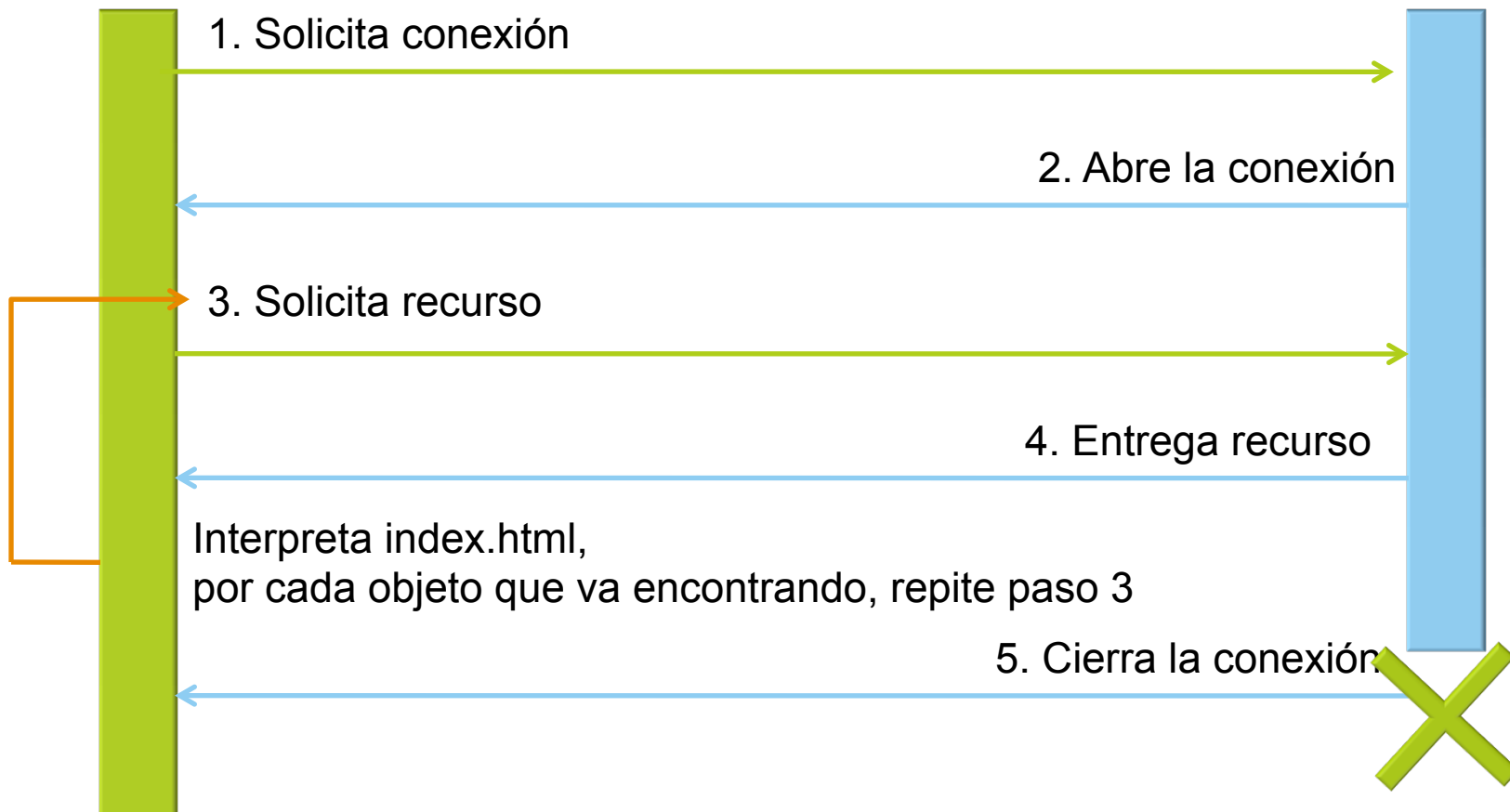


HTTP

Navegador

HTTP persistente (keep alive)

Servidor



HTTP

- Stateless
 - El servidor no mantiene información del cliente.
 - El servidor mantiene estados de las peticiones.
- Tipos de mensajes del protocolo
 - GET
 - POST
 - HEAD
 - PUT
 - DELETE
- El servidor o la aplicación receptora deben de tener una implementación de respuesta.

HTTP

- ▣ Respuestas del servidor:
 - ▣ 1xx: informativo
 - ▣ 2xx: exitoso
 - ▣ 3xx: redireccionamiento
 - ▣ 4xx: error
 - ▣ 5xx: error del servidor

Tarea 1-A

- Equipos de 2 personas
 - Crear un cliente en Java con HttpURLConnection
- Enviar peticiones de búsqueda a Google con diferentes parámetros
- Usar tanto POST como GET
- Entregar:
 - Código fuente

Tarea 1-B

- Equipos de 2 personas
 - Crear un servidor HTTP en Java, usando ***com.sun.net.httpserver***
- Se ejecuta desde 127.0.0.1 puerto 8085
- Solamente tiene un recurso, responde siempre con una imagen (.png o .jpg)
- Entregar:
 - Código fuente