

Introducción a la Arquitectura Web

IAW ANDEL 23-24

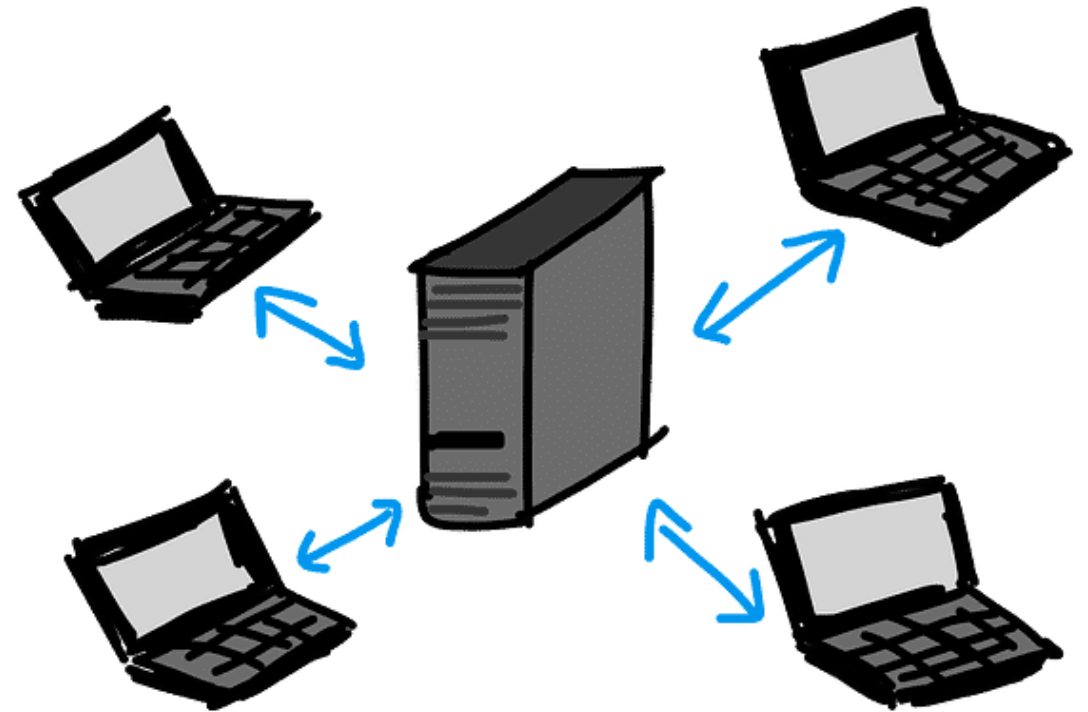
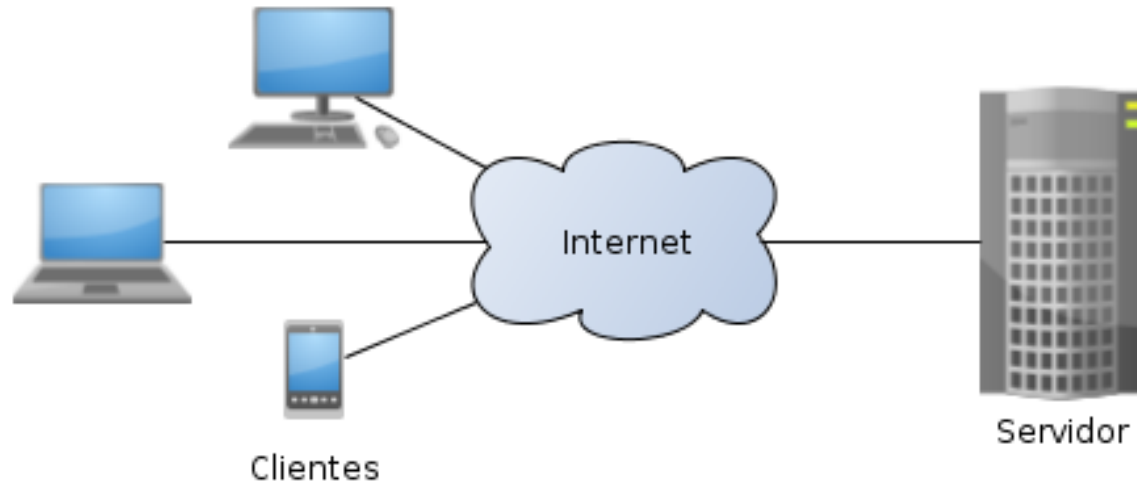
Clase 15/09

- CONCEPTO CLIENTE - SERVIDOR
- HTTP/HTTPS
- FUNCIONES HTTP
- RESPUESTAS PETICIONES

EJERCICIO VISUALIZAR LAS PETICIONES DE UN NAVEGADOR A UN SERVIDOR

CREACIÓN DE UN SERVIDOR WEB EN NODEJS

Arquitectura Cliente-Servidor



CLIENTES

- ENVÍA PETICIONES Y ESPERA RESPUESTAS
- NAVEGADOR WEB

F12 >> RED

SERVIDORES

- PROVEEDOR DE RECURSOS O SERVICIOS
- SERVIDOR WEB: PROVEEDOR DE FICHEROS
- SERVIDOR APLICACIONES: EJECUTA PETICIONES Y DEVUELVE GENERADO
- SERVIDOR DE BBDD: PETICIONES DE DATOS Y DEVUELVE INFORMACIÓN

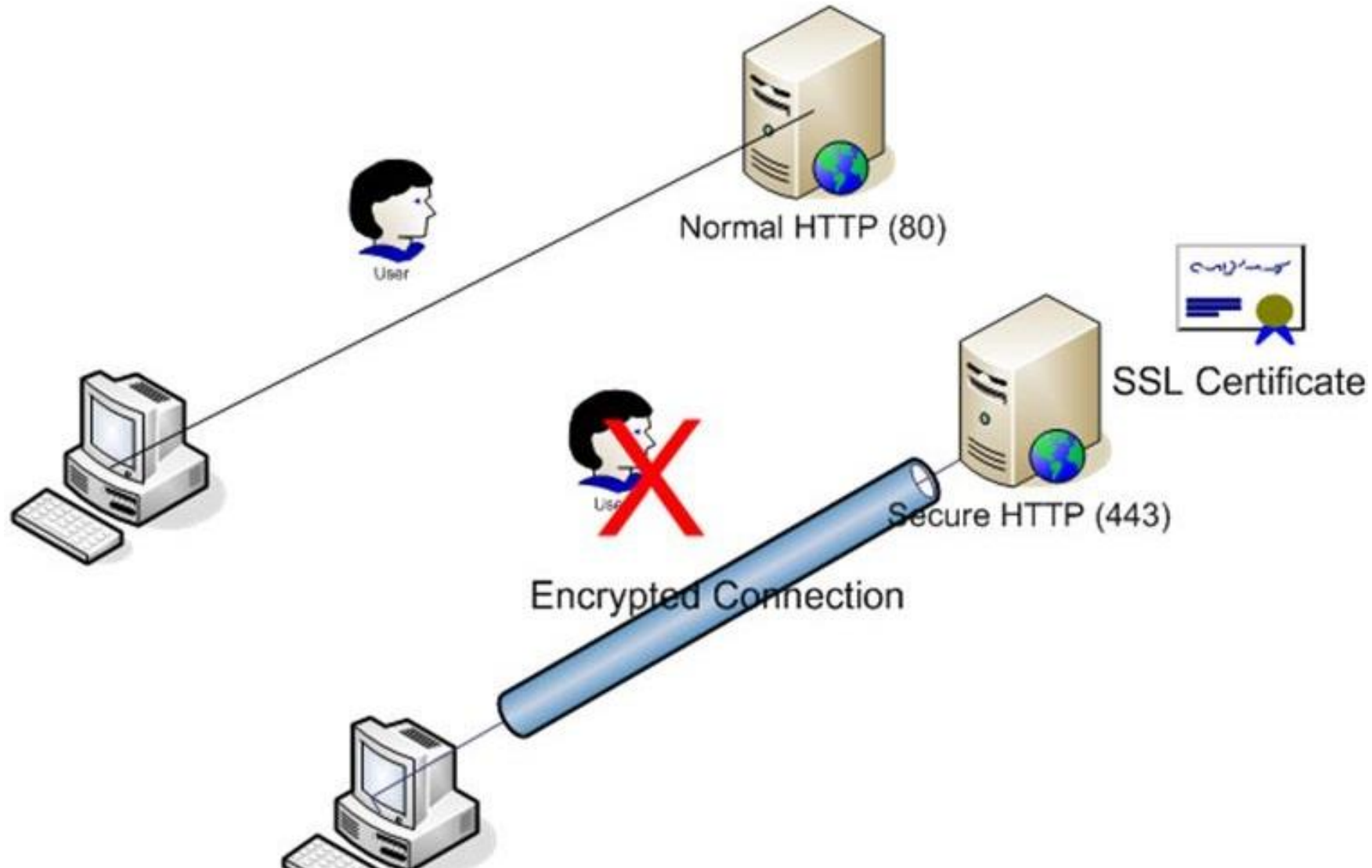
SERVIDOR

- ESCUCHA PETICIONES, LAS PROCESA Y ENVÍA UNA RESPUESTA

Ejemplo:

```
curl https://jsonplaceholder.typicode.com/posts/
```

HTTP vs HTTPS



Métodos de HTTP

EJEMPLOS MÉTODOS

<https://jsonplaceholder.typicode.com/>

GET

- Solicita una representación del recurso especificado.
- Debe ser usado solo para recuperar datos.

<https://jsonplaceholder.typicode.com/>

POST

- Se utiliza para enviar datos para ser procesados a un recurso.
- Los datos se incluyen en el cuerpo de la petición.

PUT

- Actualiza el recurso o crea un nuevo recurso si no existe, con los datos proporcionados.

DELETE

Elimina el recurso especificado.

Estados de Respuesta HTTP:

- **Respuestas Informativas (1xx):**
 - Ejemplo: 100 Continue.
- **b. Respuestas de Éxito (2xx):**
 - Ejemplo: 200 OK.
- **c. Redirecciones (3xx):**
 - Ejemplo: 301 Moved Permanently.
- **d. Errores del Cliente (4xx):**
 - Ejemplo: 404 Not Found.

Estados de Respuesta HTTP:

- **Respuestas Informativas (1xx):**
 - Ejemplo: 100 Continue.
- **Respuestas de Éxito (2xx):**
 - Ejemplo: 200 OK.
- **Redirecciones (3xx):**
 - Ejemplo: 301 Moved Permanently.
- **Errores del Cliente (4xx):**
 - Ejemplo: 404 Not Found.
- **Errores del Servidor (5xx):**
 - Ejemplo: 500 Internal Server Error.

Práctica final

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('¡Hola mundo!');
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Servidor corriendo en
http://localhost:${PORT}`);
});
```