

Agencia de Habilidades
para el Futuro

<Talento
Tech/>

Node.JS

Clase 7: Cómo funciona la web

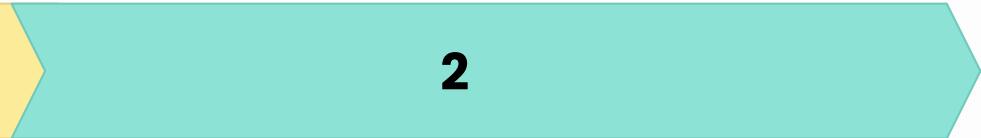
¡Les damos la bienvenida!

 Vamos a comenzar a grabar la clase.

Índice



1



2

¿Cómo funciona la web?

- ¿Cómo funciona Internet?
 - Protocolo TCP/IP
- **Modelo Cliente/Servidor**
- **Protocolo de comunicación**
- **HTTP en profundidad**

Servidores Web y patrones de Arquitectura

- **Servidores**
 - ¿Qué es un servidor?
 - ¿Cómo funciona un servidor?
 - Tipos de Servidores
 - Servidores Web
 - Alojamiento y Hostings
- **Patrones de Arquitectura**
- **MVC vs API Rest**

Objetivos de la Clase

1 Comprender el modelo cliente-servidor

Entender el concepto de cliente-servidor. Reconocer su importancia en la arquitectura de Internet. Comunicación en redes utilizando protocolos estándar

2 Identificar cómo las redes se comunican mediante protocolos

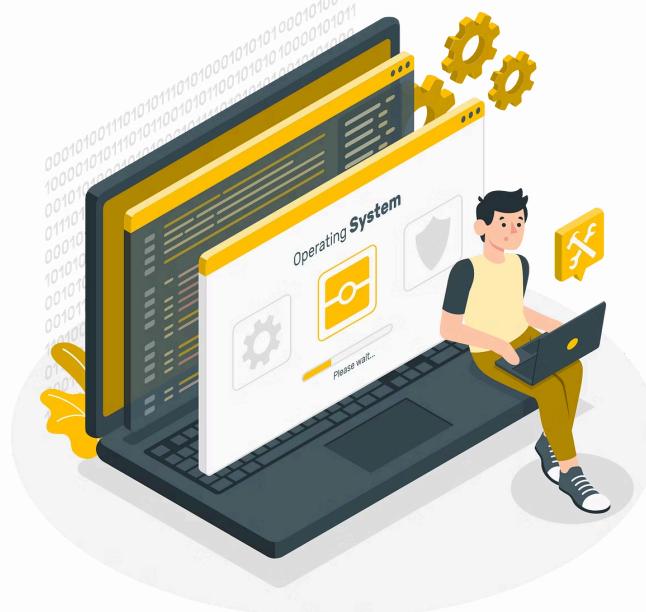
Analizar el funcionamiento del protocolo TCP/IP. Comprender el rol de TCP/IP en la transferencia de datos. Diferenciación entre HTTP y HTTPS

3 Entender las características principales de HTTP

4 Descomponer las partes de una URI (Uniform Resource Identifier). Diferenciar entre URL (Uniform Resource Locator) y URN (Uniform Resource Name)

¿Cómo funciona Internet?

¿Cómo funciona Internet?



1 Red de redes

Internet es una gigantesca "red de redes" que conecta dispositivos globalmente.

2 Idioma común

Utiliza TCP/IP como protocolo estándar para la comunicación entre dispositivos.

3 Interconexión global

Permite que todos los dispositivos estén interconectados a nivel mundial.

Protocolo TCP/IP

IP (Internet Protocol)

Encuentra y define la dirección del destino, como un código postal para dispositivos.

TCP (Transmission Control Protocol)

Garantiza que los datos lleguen completos y en el orden correcto.

Capas del Protocolo TCP/IP

Application Layer

Genera información y requiere conexiones.

Transport Layer

Maneja el envío y recepción de datos.

Internet Layer

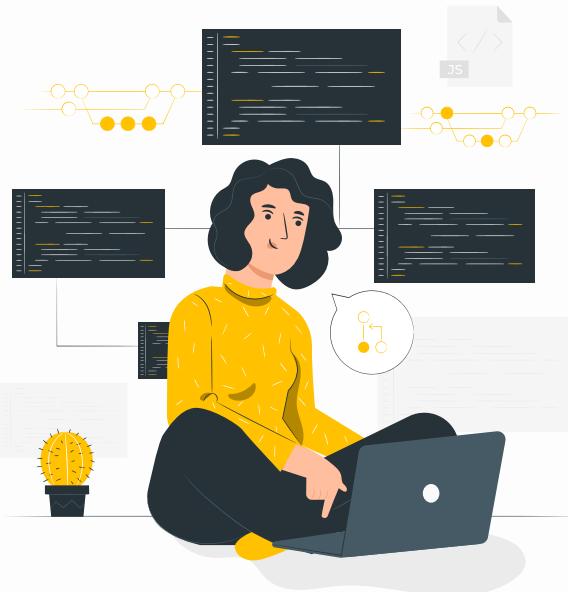
Define direcciones y rutas de datos.

Network Access Layer

Conexión física (WiFi o Ethernet).

Modelo Cliente/Servidor

Modelo Cliente/Servidor



Cliente

Solicita servicios o datos
(ej. navegador web).

Servidor

Responde a las solicitudes
y proporciona recursos.

Interacción

Intercambio de información y recursos entre dispositivos.

Protocolo de comunicación

Protocolos de comunicación: HTTP



1 Lenguaje común

HTTP es el estándar para intercambiar información en la web.

2 Sin estado

No guarda información sobre conexiones anteriores.

3 Cookies

Usadas para mantener estado en aplicaciones web.

Componentes clave de HTTP

Métodos HTTP

Acciones que el cliente solicita al servidor (GET, POST, etc.).

URLs

Identifican de manera única los recursos en la web.

Cabeceras

Contienen metadatos sobre la solicitud o respuesta.

Cuerpo del mensaje

Contenido enviado en las solicitudes o respuestas.

HTTP Seguro (HTTPS)



Encriptación

Protege los datos transmitidos entre el navegador y el servidor.

Certificado digital

Verifica la identidad del sitio y cifra la información.

Privacidad

Evita que la información pueda ser interceptada por terceros.

Certificados SSL/TLS

Verificación de identidad

Aseguran la autenticidad de un sitio web.

Encriptación de datos

Protegen la información transmitida entre usuario y servidor.

Emisión por CA

Emitidos por Autoridades de Certificación que verifican la autenticidad.



HTTP en profundidad

Métodos HTTP



GET

Solicita un recurso sin modificarlo.



POST

Envía datos al servidor.



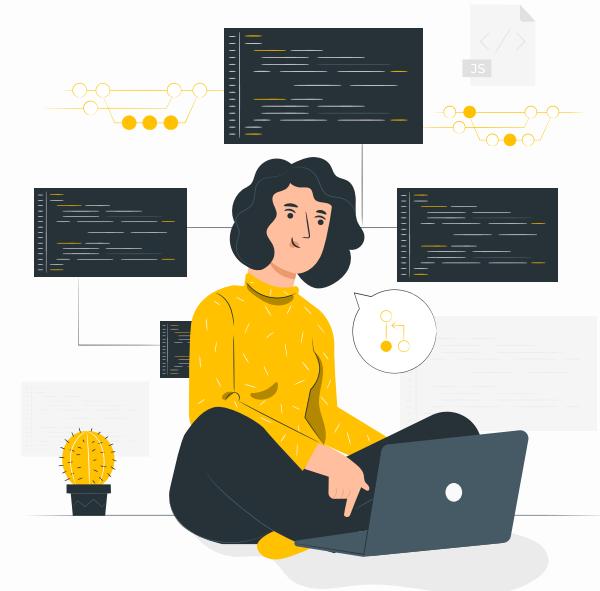
PUT

Actualiza un recurso existente completamente.



DELETE

Elimina un recurso identificado.



Headers HTTP

Cabeceras de solicitud

- User-Agent
- Accept
- Authorization
- Cookie

Cabeceras generales

- Connection
- Date
- Transfer-Encoding
- Upgrade

Cabeceras de respuesta

- Content-Type
- Set-Cookie
- Cache-Control
- Content-Length

Cabeceras específicas para control y seguridad

- Access-Control-Allow-Origin
- Strict-Transport-Security
- X-Content-Type-Options
- Content-Security-Policy

Códigos de estado HTTP

200 OK

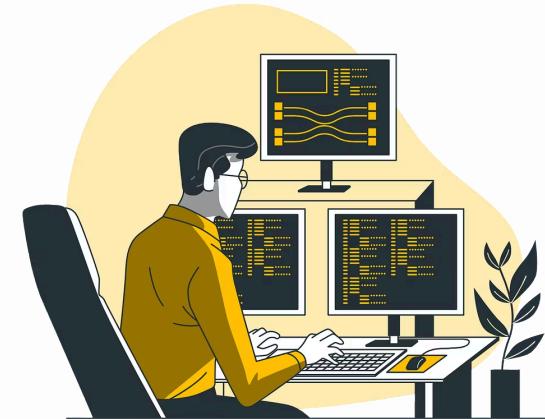
La solicitud fue procesada correctamente.

404 Not Found

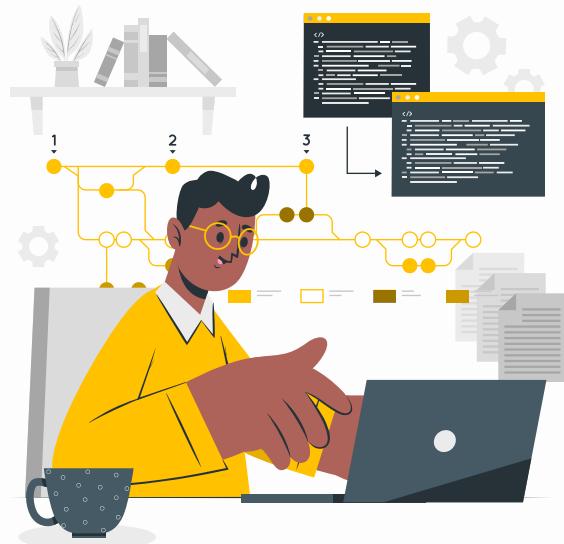
El recurso solicitado no existe.

500 Internal Server Error

Error en el servidor al procesar la solicitud.



URI: URL + URN



URL

Localizador
uniforme de
recursos.

Ejemplo:
[https://www.ejemo
plo.com](https://www.ejemplo.com)

URN

Nombre uniforme
de recursos.

Ejemplo:
[/articulos?category=tecnologia](#)

URI

Identificador
uniforme de
recursos.

Combina URL y
URN.

Conclusión

1 Fundamentos web

Comprender estos conceptos es esencial para el desarrollo web moderno.

2 Comunicación eficiente

Protocolos y estándares permiten una web interconectada y funcional.

3 Seguridad y privacidad

HTTPS y certificados SSL/TLS son cruciales en la web actual.



Materiales y Recursos Adicionales:

-
- 1** Comunicación Cliente/Servidor - [documentos de la web](#)
 - 2** Códigos de estado de respuesta HTTP - [MDN Web Docs - HTTP Status](#)
 - 3** Métodos de petición HTTP - [MDN Web Docs - HTTP Methods](#)

Próximos Pasos:

Servidores Web y Patrones de Arquitectura:

Aprenderemos sobre cómo funcionan los servidores que dan vida a internet y conoceremos los cimientos de los proyectos de programación.

Creando un Servidor Web:

Daremos nuestros primeros pasos en la creación de servidores web con Node JS.

Modelando una API Rest:

Nos iniciaremos en la metodología para crear nuestra primera API Rest.



¡Nuevo cuestionario en Campus!

- ⓘ No olvides que los cuestionarios son de carácter obligatorio para poder avanzar con la cursada.