

Summer School 2025

Introducción a Web

Vasile Andrei Popan & Sergio Morales



¿Quiénes Somos?

Vasile Andrei Popan

Soy estudiante de 2º del grado en carrera en Ingeniería Informática en la Universidad de Deusto.

He participado en el Hackathon de la UPC, así como en varios CTFs.



Puedes encontrarme o seguir mi trabajo aquí:

Linkedin: <https://es.linkedin.com/in/vasile-andrei-popan>

Github: <https://github.com/vasileandreipopan>



Sergio Morales

Soy técnico superior en desarrollo de aplicaciones web con más de un año de experiencia en LKS Next.

Actualmente estoy cursando 2º del Grado en Ingeniería informática de la Universidad de Deusto.

Me gusta programar y los macarrones.

Y... prefiero windows a Linux (es una secta)

Puedes encontrarme en:

Web: <https://sergiomorales.dev>

Linkedin: <https://es.linkedin.com/in/sergiomoralescobo>

Github: <https://github.com/sergitxin22>

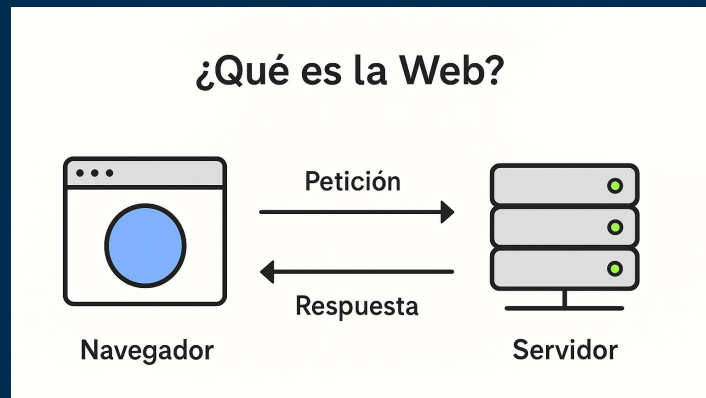
Sergio Morales & Vasile Andrei Popan



Introducción: ¿Qué es la web?

- La web es la parte visual/navegable de internet.
- **No** es lo mismo que Internet (red física).
 - Internet es la infraestructura física. Es una red global de cables, routers, antenas... que conecta dispositivos entre sí.

- Actividad: ¿Qué pasa cuando buscamos “[youtube.com](https://www.youtube.com)”? d9





Cliente/Servidor, HTTP/HTTPS

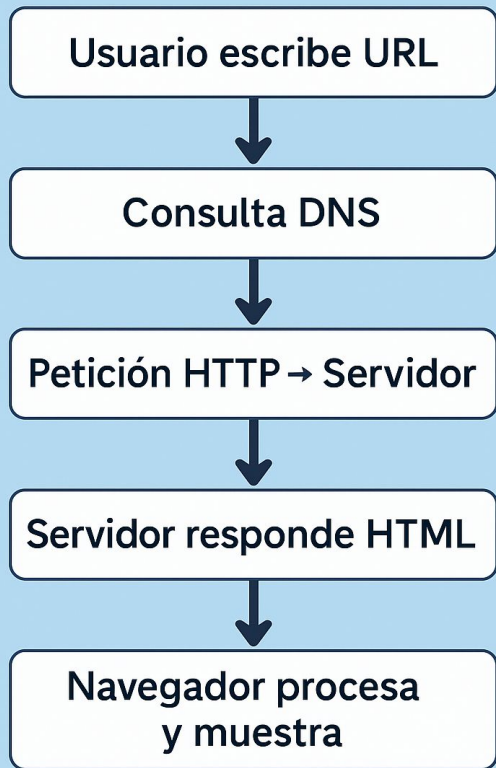
- El cliente (navegador) hace peticiones al servidor.
 - HTTP: Protocolo que define cómo se comunican los servidores.
 - HTTPS: Versión segura con cifrado de HTTP.
- Códigos de estado HTTP comunes:
 - 200: Todo funciona correctamente.
 - 304: No se ha modificado desde la última vez que se pidió porque el navegador ya tenía una copia válida en el caché.
 - 404: El servidor no encuentra el recurso que el navegador ha solicitado.
 - 500: Error interno en el servidor.
- Actividad: Usar herramientas del navegador con f12

Navegadores y procesamiento

- Recibe el HTML y lo convierte en una estructura interna llamada DOM (Document Object Model).
- Descarga el CSS externo, lo interpreta y combina con el DOM para formar el Árbol de Renderizado.
- Calcula el diseño (Layout), donde va cada elemento...
- Dibuja los pixeles en la pantalla (colores, imágenes...)
- Ejecuta el JavaScript.



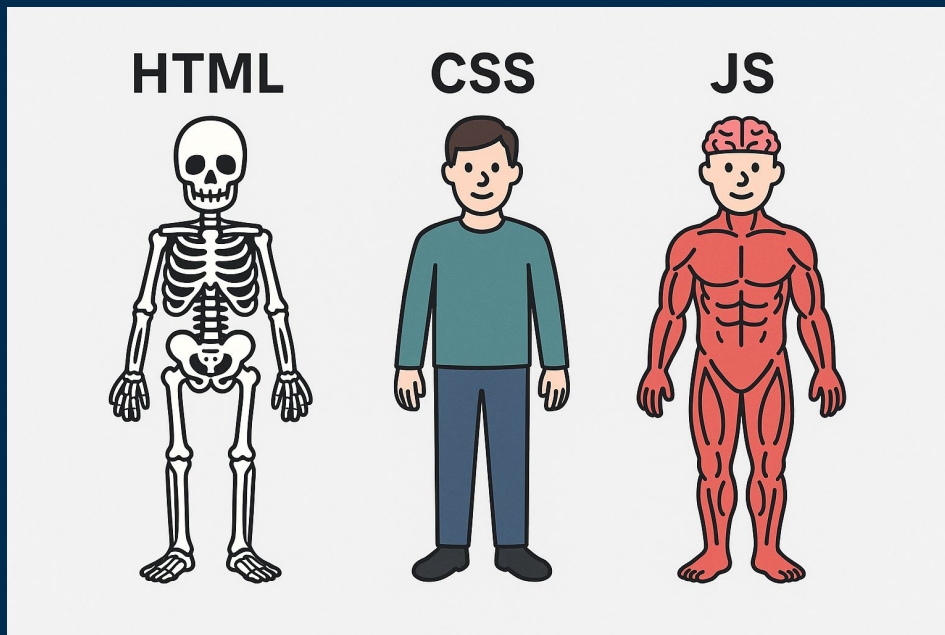
Ciclo básico de una petición web



- DNS (Domain Name Server) es como la agenda de contactos de Internet.
 - Los servidores no entienden “[google.com](https://www.google.com)”.
 - Solo entienden direcciones IP.
- Caso de HTTPS:
 - Primero se establece una conexión segura mediante el protocolo TLS (handshake).
 - Se intercambian claves de seguridad.
 - Una vez estando en ese “túnel seguro”, se hace la petición HTTP.
 - Nadie puede ver lo que estás pidiendo ni modificar el contenido que recibes.

HTML + CSS + JS

- HTML: Estructura
- CSS: Estilo
- JS: Comportamiento



<https://www.w3schools.com/>

Estructura básica de HTML

```
<!DOCTYPE html> <!-- Declara que usamos HTML5 -->
<html lang="es"> <!-- Idioma de la página -->

<head>

  <meta charset="UTF-8"> <!-- Codificación para acentos y caracteres especiales -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi primera página</title> <!-- Título que aparece en la pestaña -->

  <style>
    body { font-family: Arial, sans-serif; background-color: #f9f9f9; }
    h1 { color: steelblue; }
    p { color: #333; }
  </style>
</head>
```

Estructura básica de HTML

```
<body>
  <h1>¡Hola, mundo!</h1> <!-- Título principal -->

  <p>Este es un párrafo de texto.</p> <!-- Texto básico -->

  <a href="https://example.com" target="_blank">Visita otro sitio</a> <!-- Enlace externo -->

  <ul>
    <li>Elemento de lista 1</li>
    <li>Elemento de lista 2</li>
  </ul> <!-- Lista desordenada -->

   <!-- Imagen -->

  <input type="text" placeholder="Escribe tu nombre"> <!-- Campo de texto -->
  <button onclick="saludar()">Saludar</button> <!-- Botón que ejecuta JS -->

  <script>
    function saludar() {
      alert("¡Hola! Gracias por visitar esta página.");
    }
  </script>
</body>

</html>
```

Estilos; en línea vs interno vs externo

- Estilos en línea:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo en línea</title>
</head>
<body>

  <h1 style="color: crimson; text-align: center;">Bienvenido</h1>
  <p style="font-size: 18px; color: #444;">
    Este texto está estilizado con estilos en línea.
  </p>
  <button style="background-color: gold; padding: 10px; border: none;">
    Haz clic aquí
  </button>

</body>
</html>
```

Estilos; en línea vs interno vs externo

- Interno:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo interno</title>
  <style>
    body {
      background-color: #f4f4f4;
      font-family: Arial, sans-serif;
    }
    h1 {
      color: navy;
      text-align: center;
    }
    p {
      font-size: 18px;
      color: #333;
    }
    button {
      background-color: skyblue;
      padding: 10px;
      border: none;
    }
  </style>
</head>
<body>

  <h1>Bienvenido</h1>
  <p>Este texto está estilizado desde una hoja de estilos interna.</p>
  <button>Haz clic aquí</button>

</body>
</html>
```

Estilos; en línea vs interno vs externo

- Externo:

archivo.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo externo</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>

  <h1>Bienvenido</h1>
  <p>Este texto usa una hoja de estilos externa.</p>
  <button>Haz clic aquí</button>

</body>
</html>
```

estilos.css

```
body {
  background-color: #ffffd5;
  font-family: Verdana, sans-serif;
}

h1 {
  color: seagreen;
  text-align: center;
}

p {
  font-size: 18px;
  color: #222;
}

button {
  background-color: palegreen;
  padding: 10px 15px;
  border: 1px solid #ccc;
}
```

Comportamientos básicos con JavaScript (JS)

- Manipulación del DOM

```
document.getElementById("miElemento").innerText = "Hola Mundo";
```

- Eventos

```
<button onclick="alert('¡Hola!')">Haz clic</button>
```

- Validación de formularios

```
if (email.includes("@")) {
  console.log("Email válido");
}
```

- Funciones básicas

```
function saludar(nombre) {
  return "Hola, " + nombre;
}
```




Cookies



- Pequeños archivos de texto que guarda el navegador.
- Se usan para guardar información del usuario, como idioma o sesión.
- Se envían automáticamente con cada petición HTTP.

```
// Crear una cookie con nombre "usuario" y valor "Juan"
document.cookie = "usuario=Juan; expires=Fri, 31 Dec 2025 23:59:59 UTC;";

// Ejemplo petición con la cookie
fetch('https://example.com/api/data', {
  method: 'GET',
  credentials: 'include', // Asegura que se envíen cookies
  headers: {
    'Content-Type': 'application/json'
  }
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

LocalStorage

- Almacena datos localmente sin fecha de expiración.
- Persiste incluso si cierras el navegador.

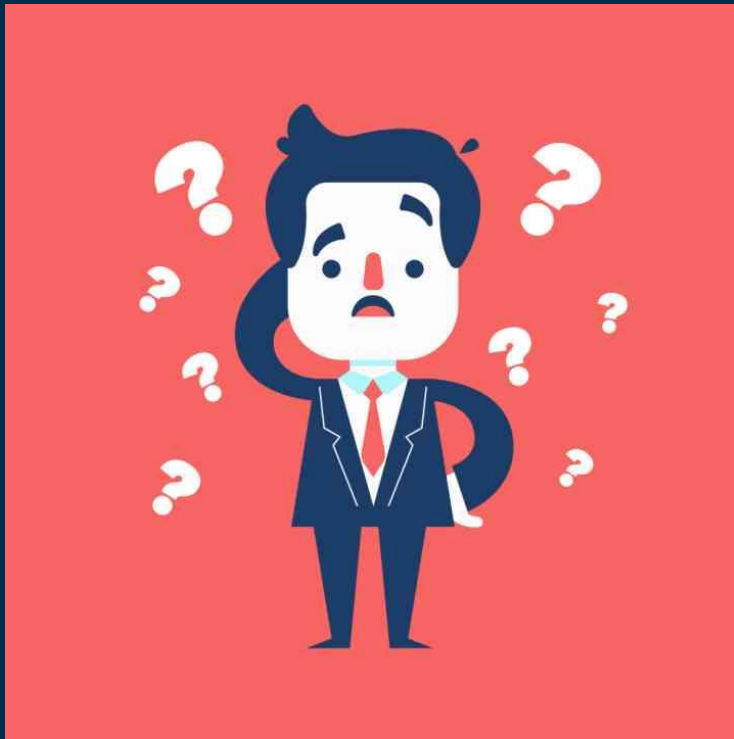
```
localStorage.setItem("tema", "oscuro");  
let tema = localStorage.getItem("tema");
```

SessionStorage

- Similar a LocalStorage, pero se borra al cerrar la pestaña o navegador.

```
sessionStorage.setItem("usuarioTemporal", "Ana");  
let usuarioTemporal = sessionStorage.getItem("usuarioTemporal");
```

Preguntas/aportaciones



Summer School 2025

Muchas gracias

Descubre **comunidad**, descubre **0xDecode**

