INSTITUTO KHIPU

Semestre IV

Sesión 25

PROGRAMA DE ESTUDIOS

DESARROLLOS DE SISTEMAS DE INFORMACIÓN

DESARROLLO FRONTEND DE SISTEMAS DE WEB

Tema

ALMACENAMIENTO EN EL NAVEGADOR (COOKIES Y LOCALSTORAGE)

Almacenamiento en el Navegador (Cookies y LocalStorage)

1. Parte Teórica

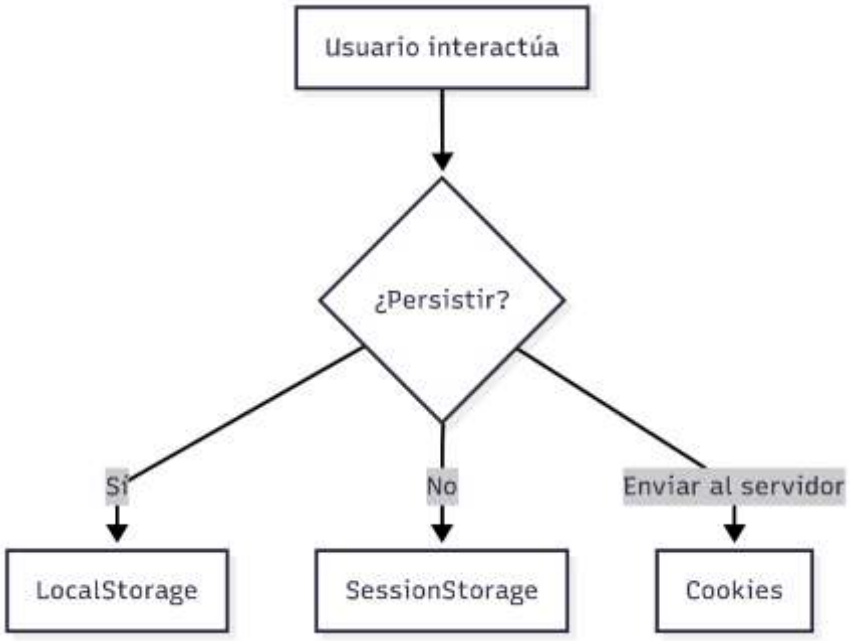
Introducción

El almacenamiento en el navegador permite guardar datos directamente en el cliente (navegador web) para mejorar la experiencia del usuario. Esto incluye:

- Persistencia de datos (ej: preferencias de usuario).
- Reducción de peticiones al servidor (mejor rendimiento).
- Personalización (ej: tema oscuro, idioma).

Definición: Mecanismos que permiten guardar datos en el navegador del usuario para mejorar la experiencia (ej: recordar preferencias, mantener sesiones).
Importancia: Personalización, persistencia de datos y reducción de solicitudes al servidor.

Flujo de Almacenamiento:



Tipos de Almacenamiento en el Navegador

Cookies

- **¿Qué son?** Pequeños archivos de texto (máx. ~4KB) que el navegador almacena y envía al servidor en cada petición HTTP.
- **Uso común:** Autenticación (tokens), preferencias de usuario (idioma).
- **Caducidad:** Se definen con una fecha de expiración (expires).

Ejemplo de creación en JavaScript:

document.cookie = "usuario=Carlos; expires=Fri, 31 Dec 2025 12:00:00 UTC; path=/";

Lectura: (javascript)

console.log(document.cookie); // "usuario=Carlos; tema=oscuro"

LocalStorage

¿Qué es? Almacenamiento persistente (hasta que el usuario lo borre) con mayor capacidad (~5-10MB).

Uso común: Guardar preferencias (tema, datos de formularios).

Métodos clave:

Método	Descripción
localStorage.setItem(key, value)	Guarda un dato (key: value).
localStorage.getItem(key)	Recupera un valor.
localStorage.removeItem(key)	Elimina un dato específico.
localStorage.clear()	Borra todos los datos.

Ejemplo:

```
// Guardar tema preferido
localStorage.setItem("tema", "oscuro");

// Recuperar tema al cargar la página
```

```
const tema = localStorage.getItem("tema") || "claro";
document.body.className = tema;
```

Ventajas:

✅ Mayor capacidad que las Cookies.

✅ Fácil de usar (API sencilla).

Desventajas:

❌ No se envían automáticamente al servidor.

❌ Solo almacena strings (usar JSON.stringify para objetos).

SessionStorage

¿Qué es? Similar a LocalStorage, pero los datos se borran al cerrar la pestaña/navegador.

Uso común: Datos temporales (ej: carrito de compras en una sesión).

Ejemplo:

```
sessionStorage.setItem("carrito", JSON.stringify(["producto1", "producto2"]));

// Recuperar datos
const carrito = JSON.parse(sessionStorage.getItem("carrito")) || [];
```

Diferencias clave vs LocalStorage:

Característica	LocalStorage	SessionStorage
Persistencia	Hasta borrado manual	Solo en la sesión actual
Alcance	Mismo dominio	Misma pestaña

Cuándo Usar Cada Uno?

Caso de Uso	Recomendación
Autenticación (JWT, tokens)	Cookies (seguras con HttpOnly)
Preferencias de usuario (tema, idioma)	LocalStorage

Datos temporales (carrito)	SessionStorage

En resumen:

Característica	Cookies	LocalStorage	SessionStorage
Persistencia	Caducidad definida (manual o automática)	Persistente hasta ser borrado	Solo durante la sesión actual
Capacidad	~4KB	~5-10MB	~5-10MB
Acceso	Enviadas en cada petición HTTP	Solo en el cliente	Solo en el cliente
Uso común	Autenticación, preferencias	Datos no sensibles (ej: tema)	Datos temporales (ej: carrito)

Cookies en JavaScript

- **Creación:** `document.cookie = "username=John; expires=Thu, 18 Dec 2025 12:00:00 UTC; path=/";`
- **Lectura:** `console.log(document.cookie); // Muestra todas las cookies`
- **Eliminación:** Establecer fecha de expiración pasada.

LocalStorage y SessionStorage

- **Métodos clave:**
```
localStorage.setItem("clave", "valor"); // Guardar
localStorage.getItem("clave");          // Leer
localStorage.removeItem("clave");       // Eliminar
localStorage.clear();                   // Limpiar todo
```
- **SessionStorage:** Mismos métodos, pero con `sessionStorage`.

2. Parte Práctica

2.1 Ejercicio Guiado

Objetivo: Crear una página que recuerde el nombre del usuario con LocalStorage.

```
<!DOCTYPE html>
<html>
```

```
<body>
  <input type="text" id="username" placeholder="Ingresa tu nombre">
  <button onclick="guardarNombre()">Guardar</button>
  <script>
    function guardarNombre() {
      const nombre = document.getElementById("username").value;
      localStorage.setItem("nombreUsuario", nombre);
      alert("Nombre guardado!");
    }
    // Al cargar la página:
    window.onload = function() {
      const nombreGuardado = localStorage.getItem("nombreUsuario");
      if (nombreGuardado) {
        document.getElementById("username").value = nombreGuardado;
      }
    };
  </script>
</body>
</html>
```

2.2 Actividad Individual

Tarea: Implementar un selector de tema (claro/oscuro) usando LocalStorage.

```
// Ejemplo de solución:
function cambiarTema(tema) {
  localStorage.setItem("tema", tema);
  document.body.className = tema;
}
// Al cargar la página:
document.body.className = localStorage.getItem("tema") || "claro";
```

3. Ejercicios y Tareas

3.1 Ejercicio en Clase

- **Desafío:** Usar Cookies para recordar el idioma preferido. `document.cookie = "idioma=es; expires=Fri, 31 Dec 2025 12:00:00 UTC; path=/";`

3.2 Tarea para la Casa

- **Consigna:** Crear una página que:
- Almacene el nombre y color de fondo en LocalStorage.
- Use Cookies para recordar el tamaño de fuente preferido.
- Incluya botones para resetear los valores.

Código de referencia:

```
<input type="color" id="colorFondo">
<button onclick="guardarPreferencias()">Guardar</button>
<script>
  function guardarPreferencias() {
    const color = document.getElementById("colorFondo").value;
    localStorage.setItem("colorFondo", color);
    document.cookie = "fontSize=16px; expires=Fri, 31 Dec 2025 12:00:00 UTC";
  }
</script>
```

5. Recursos Adicionales

- **Vídeo:** Diferencias entre Cookies y LocalStorage
- **Artículo:** Seguridad en el almacenamiento web

INSTITUTO

KHIPU